haking9 +CD

hakin9.live – tutoriales y documentación, no requiere instalación
Shadow Database Scanners + licencia de 30 días para 2 direcciones IP
versión de 90 días de **Outpost PRO Firewall 3.51**





¿cómo defenderse?

Hard Core IT Security Magazine

Nº 17 Precio 7,50 € ISSN: 1731-2930, Bimestral

Network Defense

Victor Oppleman muestra los secretos de los avanzados ataques DDoS



Ingeniería Inversa: Desensambladores de tamaño
Escribimos aplicaciones al análisis de Malware

Problemas con autenticación HTTP
Las vulnerabilidades del metodo Basic

Análisis de tráfico en la Red
Herramientas y técnicas para detectar los ataques

PARA PRINCIPIANTES

Know-how – Protección de IPv6
Todo lo que debes saber sobre IPv6

Ingeniería social
Un ataque a tu cerebro



EN CD

Shadow Database Scanners + licencia de 30 días para 2 dirreciones IP

Outpost PRO Firewall 3.51 versión de 90 días

+ 23 tutoriales

incluyendo 4 nuevos: • Problemas con autenticación HTTP • Análisis de tráfico en la Red

NUEVOS E-BOOKS: Linux IPv6 HOWTO • Securing Debian Manual • Snort Users Manual • SQL Injection Protection

00017
8 414090 030076

Ataques DDoS • Protección de IPv6 • Ingeniería social • Ingeniería Inversa: Desensambladores de tamaño • Problemas con autenticación HTTP

haking Nº 17



La mejor distribución de Linux

Aurox...
porque funciona

Aurox. Mejor soporte para el hardware

La distribución completa de Linux

basada en Fedora Core 4

Contiene **Aurox Live** sistema que arranca directo desde el DVD

2000 paquetes de software de usuario
Mejor soporte para el hardware (configuración automática de dispositivos móviles)
Estabilidad (el sistema testado por grupos independientes de testers)
Soluciones de escritorio cómodas (KDE, GNOME, XFCE)
Aplicaciones multimedia (Audio-¡editarás cada fichero de sonido!, Vídeo –¡verás cada película!)

Configuración automática de tarjetas WiFi la posibilidad de aprovechar los drivers de Windows
Soporte para los modems ADSL
Open Clip Art Library Una librería con más de 450 gráficos para el uso de oficina
KDE 3.4.2 El entorno gráfico estable más reciente
OpenOffice 2.0 Paquete de oficina compatible con Microsoft Office
Juegos 3D Cube y TribalTrouble







Redactor
Marek Bettman

¿Virus o no virus?

Hace poco el mundo recibió el mensaje sobre la creación del primero virus *que funciona* tanto en las versiones de Windows como en las versiones de Linux. Recordar que (citando Wikipedia) – *El virus informático es sobre todo una aplicación informática simple que de manera intencionada se copia sin la autorización del usuario [...] requiere portador en forma de la aplicación informática, correo electrónico, etc. [...].* ¿Cómo, con esta información entender la descripción de la cual resulta que para activar un virus en Linux se requiere del usuario la descarga del archivo, compilarlo o ejecutar el archivo binario? ¿Por qué *Kaspersky Lab* – en el que trabajan especialistas del sector – con tanto gusto empaquetó con tags la maliciosa aplicación con el nombre de virus informático de plataforma múltiple y lo nombraron *Virus.Linux.Bi.a/Virus.Win32.Bi.a*, a pesar de que a primera vista no es un virus? ¿Por qué o bien, sobre todo está seguro que para que el nuevo virus funcione en los kernel más reciente de la serie 2.6 se requería el parche escrito por Linus Torvalds? ¿Por qué la aparición de esta aplicación fue popularizada como el final de la era de los sistemas seguros del pingüino a pesar de que es solamente POC (*Proof of Concept*) y ya antes surgieron virus para Linux (actualmente hay 40), MacOS (unas decenas) y sistemas comerciales de UNIX (unos)? ¿Por fin – los hechos tienen algo en común con la voluntad de entrar de las grandes empresas que crean aplicaciones para el mercado de los sistemas de Linux?

La opinión sobre la completa resistencia de Linux a los virus es errónea y esto se sabe desde hace mucho. Además, a medida que se populariza este sistema no solamente en los entornos de producción sino que también en los ordenadores de casa, la probabilidad de que aparezcan aplicaciones maliciosas se acerca a 1. El Linux de escritorio con Firefox y Thunderbird nunca *conseguirá* tal sensibilidad a todo tipo de virus como Windows con Internet Explorer y Outlook.

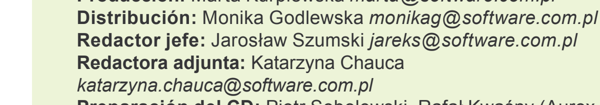
A pesar de todo me sorprende el comportamiento de *Kaspersky Lab*. Me interesa saber ¿cuándo empezará la promoción agresiva (de las ya existentes) conjunto de aplicaciones antivirus para Linux?

Me viene a la cabeza una explicación más de este fenómeno: *Microsoft* antes de introducir en el mercado *Windows Vista* quiere desacreditar a Linux, como sistema seguro y convencer a las personas que piensan en migrarse al sistema gratuito de la misma sensibilidad a los peligros que vienen de la red. Tal opinión seguramente *ayudaría* en tomar la decisión de comprar una nueva versión del sistema ya empelado antes que migrar a uno nuevo igual de difícil y susceptible a los ataques del sistema...

Es necesario recordar una cosa – incluso las aplicaciones más avanzadas antivirus no piensan en los usuarios y administradores. Lo más importante es la conciencia de la amenaza y saber dónde buscar información sobre los métodos de defensa y prevención. Tales mensajes independientemente del grado de peligro del *Virus.Linux.Bi.a/Virus.Win32.Bi.a* – se suministran y suministrarán en la revista *hakin9*.

¡Os invito a leerla!

Marek Bettman

haking
está editado por Software-Wydawnictwo Sp. z o.o.
Dirección: Software-Wydawnictwo Sp. z o.o.
ul. Piaskowa 3, 01-067 Varsovia, Polonia
Tfno: +48 22 887 10 10, Fax: +48 22 887 10 11
www.hakin9.org
Producción: Marta Kurpiewska *marta@software.com.pl*
Distribución: Monika Godlewska *monikag@software.com.pl*
Redactor jefe: Jarosław Szumski *jareks@software.com.pl*
Redactora adjunta: Katarzyna Chauca
katarzyna.chauca@software.com.pl

Preparación del CD: Piotr Sobolewski, Rafał Kwaśny (Aurox Core Team)
Composición: Anna Osiecka *annao@software.com.pl*
Traducción: Osiris Pimentel Cobas, Małgorzata Janerka, Hanna Grafik-Krzymińska, Mariusz Muszak, Paulina Stosik, Raúl Nanclares, Tanie Tłumaczenia.
Corrección: Jesús Alvárez Rodrígez, Jorge Barrio Alfonso, Alfonso Huergo Carril
Betatesters: Juan Pérez Moya, Jose M. García Alias, Luis Peralta Nieto, Jose Luis Herrera, Paco Galán

Publicidad: *adv@software.com.pl*
Suscripción: *suscripcion@software.com.pl*
Diseño portada: Agnieszka Marchocka

Las personas interesadas en cooperación rogamos se contacten: *cooperation@software.com.pl*

Si estás interesado en comprar la licencia para editar nuestras revistas contáctanos:
Monika Godlewska
e-mail: *monikag@software.com.pl*
tel.: +48 22 887 12 66
fax: +48 22 887 10 11

Imprenta: 101 Studio, Firma Tęgi
Distribuye: coedis, s.l.
Avd. Barcelona, 225
08750 Molins de Rei (Barcelona), España

La Redacción usa el sistema de composición automática AUPUS
Los diagramas han sido elaborados con el programa smartdraw.com
de la empresa SmartDraw
El CD incluido en la revista ha sido comprobado con el programa AntiVirenKit, producto de la empresa G Data Software Sp. z o.o.

La revista haking es editada en 7 idiomas:



Advertencia

¡Las técnicas presentadas en los artículos se pueden usar SÓLO para realizar los tests de sus propias redes de ordenadores! La Redacción no responde del uso inadecuado de las técnicas descritas. ¡El uso de las técnicas presentadas puede provocar la pérdida de datos!



Agujeroagujeros en MacOS X

Tom Ferris, especialista de las investigaciones de la seguridad de los sistemas informáticos, descubrió los detalles relacionados con los agujeroagujeros descubiertos en el sistema operativo *Mac OS X*. El más importante de los agujeroagujeros aparece en el navegador Safari. Gracias a ella el atacante puede ejecutar por medio del navegador cualquier código u ocasionar su avería. Otro agujero podemos emplear por medio de un especialmente preparado archivo gráfico en el formato *TIFF, BMP* ó *GIF*. El ataque eficaz conduce a la avería de la aplicación que sirvió para abrir el archivo preparado. El siguiente error aparece en la forma de servicio de los archivos empaquetados. Permite llevar a la avería de la aplicación o bien a la realización de cualquier código en la máquina atacada.

El primero de los agujeroagujeros fue evaluado por *Ferris* como amenazador, los demás como medianamente amenazadores.

Las organizaciones especialistas que se ocupan de la seguridad de ordenadores tales como SANS Internet Storm Centre, evalúan la amenaza como altamente crítica, ya que los agujeroagujeros permiten ejecutar de forma remota cualquier código o realizar el ataque DoS.

Ataque a los virus

Los especialistas que se ocupan de la seguridad informática advirtieron a los internautas sobre un correo que viaja por Internet haciendo de correo real de uno de los fabricantes de aplicaciones antivirus, constituye una seria amenaza para los sistemas informáticos.

El falso correo electrónico en realidad sirve a los chive criminales para imposibilitar la actualización de las aplicaciones antivirus instaladas en el ordenador atacado.

En el contenido del correo se incluyó la información de que la máquina del remitente está infectada con el virus *w32.aplore@mm*. Allí se encuentra un enlace que supuestamente elimina el virus. Al hacer clic en el código iniciamos tales cambios de la configuración del ordenador que las aplicaciones antivirus dejan de actualizarse.

El correo falso fue descubierto primero en Asia, desde donde la información llegó a Europa y América.

Agujeroagujeros en SSH y SSL/TSL

Los especialistas de la *Universidad Politécnica de Wrocław* (*Polonia*) que trabajan bajo la tutela del profesor doctor Mirosław Kutyłowski descubrieron puntos débiles en los protocolos SSL/TSL y SSH, los protocolos más populares que garantizan la comunicación segura en Internet.

En caso de emplear el mecanismo descubierto por el virus que infecta a las respectivas aplicaciones de usuario, es posible descifrar todos los mensajes enviados y recibidos por el usuario con el empleo de estos protocolos.

La novedad del ataque es asegurar un cierto monopolio: incluso una información completa sobre el virus (y el material criptográfico que incluye) no permite realizar un ataque. Además, necesitamos claves criptográficas adicionales conocidas solamente por los constructores del virus. Además, la aplicación infectada se comporta de manera que no se diferencia de la correcta.

La sensibilidad descubierta de estos protocolos es muy importante, teniendo en cuenta el potencial espionaje económico, acceso no autorizado a los bancos electrónicos de Internet, etc. Esto muestra lo importante que es la cuestión de la confianza completa del fabricante de aplicaciones suministradas sin códigos fuente.

Al mismo tiempo se elaboraron unas pequeñas modificaciones de los estándares que permiten eliminar las amenazas descritas. Sobre la descripción de esta informaron los órganos de seguridad, pronto se propondrá el parche para el estándar internacional de SSL.

China, piratería y Microsoft

Los ordenadores fabricados por los fabricantes chinos deben tener instalado un sistema operativo antes de que abandonen la fábrica – es un decreto oficial de las autoridades de Beijing. Esto tiene como objetivo solucionar el problema de la piratería en China y suavizar el conflicto con EE.UU.

China es un paraíso para los vendedores de aplicaciones pirata. Según informa la agencia *Reuters*, la versión pirata de *Windows XP Professional* se puede comprar aquí por 30 yuanes. La versión original cuesta unos 2 000 yuanes (249 dólares).

Se estima que en China de cada diez ordenadores nueve tienen copias ilegales. Y esto no gusta a EE.UU. de donde proceden la mayoría de aplicaciones.

El decreto sobre la obligación de la instalación en fábrica del sistema operativo debe demostrar que China piensa seriamente en la lucha contra la piratería. Pero no solamente esto – la acción masiva de reemplazo de las aplicaciones piratas con las legales en los ordenadores de las oficinas les costó a las autoridades de Beijing unos 17,5 millones de dólares.

Como podemos observar, *Microsoft* se aprovecha de la lucha contra la piratería – esta semana oficialmente ha firmado contratos con los fabricantes de tres ordenadores de venta del sistema Windows. Se preve que el consorcio ganará hasta 1,6 mil millones de dólares.



Figura 1. *Sitio web de Microsoft España*

Estafadores de subastas

En el servicio alemán de subastas *http://www.mobile.de*, un cliente encontró un *Land Rover* del año 1998 en estado ideal, al precio de 2,900 euros (incluyendo el coste de transporte). El coche lo vendía una tal Alexa Mangel. En los correos escribió sobre su divorcio y mudanza a Gran Bretaña. Explicó que había comprado el coche en Alemania, y que para ella es un problema, ya que el volante está a la izquierda y no a la derecha. La transacción iba a realizarse por medio de la empresa *International Cargo Spedition* (ICS).

El escenario era el siguiente: Mangel suministra el coche al intermediario (ICS) y paga el coste de envío. Cuando la empresa reciba el coche, se pone en contacto con el comprador y le informa que dentro de las 24 horas éste debe pagarlo.

Por el momento todo parece bien – al comprar por Internet los objetos de gran valor muchas veces empleamos servicios de fideicomiso (el más popular es *Escrow.com* que pertenece a *eBay*). El cliente paga el dinero a la cuenta bancaria de tal empresa y el vendedor envía la mercancía. Cuando el comprador confirme que lo ha recibido, el dinero llega a la cuenta del vendedor. Sin embargo, aquí el comprador iba a pagar con antelación al contado por medio de la red *Western Union* al nombre privado de uno de los agentes ICS de Londres.

El vendedor subrayó que el cliente tenía 10 días para probar el coche. Si hay algo que no lo guste, puede devolverlo sin coste adicional. Incluso envió el listado de puntos en el domicilio del cliente donde éste pudiera efectuar la transferencia.

Unos días más tarde el cliente recibió un correo de ICS con el número de envío. Pudo seguirlo por medio de Internet, pero seguía desconfiando – introdujo el nombre de la empresa en el buscador de Internet. Resultó que sobre ICS se habló mucho en los foros que advierten sobre estafadores (entre otros en los Países Bajos, Alemania y República Checa). La empresa aparecía bajo diferentes nombres, entre otros *World Shipping, International Cargo Spedition*, *Euro Parcel Distribution* etc. Cada semana cambiaba también de dirección Web – se encontraron unas decenas, entre otras: *autoscoutmarket.com*, *europe-transcontinentals.com*. Todas eran sitios preparados de manera profesional que podían ser modelos de los sitios de empresas reales (actualmente todas ya no son accesibles). Los estafadores pocas veces cambian los datos de contacto – número de teléfono (que no funciona) y dirección de la sede en muchos casos es la misma.

El mecanismo del engaño es simple – no hay ningún coche, el vendedor y la empresa de fideicomiso son ficticios y sirven para estafar dinero.

Los falsos agentes de fideicomiso aparecieron en 2002. En toda Europa se describieron los casos de las personas que habían perdido miles de euros por su culpa.

Religión contra pornografía

En Israel un grupo de crackers ortodoxas declaró la guerra a las páginas pornográficas en Internet. Su actividad es cambiar el contenido de servicios – en vez de las fotos desnudas los usuarios de las páginas encontrarán la imagen de un estimable rabí.

Por el momento – según informó el diario Yedioth Aharonot – un grupo de crackers atacó solamente las páginas israelíes. En un caso se eliminó el contenido de toda una página pornográfica. Los intrusos ortodoxos firman todas las páginas que atacan. Debajo de la foto de un rabí de barba cana Menahem Mendel Schneerson aparece la leyenda: *Nosotros, el grupo Da-Net, atacamos este sitio y eliminamos todas estas asquerosidades*.

Luego, los crackers anaden que las páginas pornográficas llevaron a mucha gente a problemas, desgracias e incluso a la muerte.

Troyano chantajista

El troyano que acaba de crearse bloquea el ordenador infectado y de su titular demanda dinero por el desbloqueo. El código nocivo llamado *Ransom.A* crea en el disco numerosos archivos .exe, y, luego, informa: Los archivos eliminados se guardarán en el directorio escondido y se recuperarán durante el proceso de desinstalación. El troyano informa que cada 30 minutos eliminará un archivo.

En la pantalla del ordenador infectado se muestran imágenes pornográficas e información que dice que con cada reinicio del ordenador aparecerán en el disco las siguientes copias del troyano y se eliminarán los siguientes archivos importantes. Después de mostrar el Administrador de Tareas el usuario de la máquina verá numerosos procesos ejecutados por el troyano. Una prueba de desactivar uno de ellos terminará con mostrar la imagen y la leyenda *No morimos. Nos multiplicamos. ¿Ctrl+Alt+Del hoy no funciona algo, verdad?*.

Los creadores del código nocivo dicen que pueden eliminarlo solamente de una manera. Es necesario enviar por medio del servicio *Western Union* la cuota de 10,99 USD y en lugar del usuario introducir "4870930101308697". Después de pagar la cuota demandada recibiremos la confirmación en la cual se encontrará el número de identificación. Será necesario introducir este número en el ordenador infectado y el troyano se desinstalará.

Oracle introduce parches

Oracle publicó la siguiente actualización sumatoria de la seguridad de aplicaciones (*Oracle Critical Patch Update*). La actualización incluye parches en 36 productos, incluyendo: Oracle Database, Oracle Application Server, Oracle Collaboration Suite, Oracle E-Business Suite and Applications, Oracle Pharmaceutical Applications y Oracle Enterprise Manager.

Además, se publicó la versión actualizada de la herramienta para comprobar las contraseñas predeterminadas de los productos Oracle (Oracle Default Password Scanner).

Más información podemos encontrar en la página oficial *http://www.oracle.com/technology/deploy/security/pdf/cpuapr2006.html*

Internet Explorer otra vez peligroso

Michał "lcamtuf" Zalewski detectó en el navegador Internet Explorer un agujero serio que probablemente permite realizar en él cualquier código.

El agujero está relacionado con el soporte de los tags anidados OBJECT. Por medio de la carga de la página preparada respectivamente en el navegador podemos forzar la administración de la memoria. Es probable que se pueda seleccionar el código HTML de tal forma que el sistema realice la serie ordenada de instrucción.

Para emplear el agujero es necesario convencer al usuario de que abra la página preparada en el navegador. Entonces, es necesario recordar absolutamente de no visitar las páginas Web en las que no confiamos completamente y no ejecutar enlaces de fuentes inseguras (por ejemplo, recibidas por medio del mensajero o mensaje e-mail). Podemos también emplear uno de los navegadores Web alternativos, por ejemplo, Opera ó Mozilla Firefox.

Primer spamer australiano en la cárcel

La organización australiana ACMA (*Australian Communications & Media Authority*) informó que era capaz de llevar a condenar la primera persona acusada a partir de la ley australiana contra spam *Australian Spam Act*.

Saine Mansfield fue considerado culpable de enviar 56 millones de correos electrónicos no deseados. No eran suficientes las explicaciones del acusado que los usuarios del correo electrónico aceptaron recibir los correos. Mansfield trató de argumentar que el listado de direcciones a las cuales trató de enviar spam, fue creado antes de la entrada en vigor de *Australian Spam Act*. *El hecho de que las direcciones puedan recogerse antes de que esté prohibido no significa que la ley es válida desde el momento de su entrada en vigor* – dijo el juez Nicholson.

El Tribunal Federal todavía no anunció la pena.

Microsoft contra la Comisión Europea

Después de siete años de combates legales empieza la lucha decisiva de *Microsoft* contra la *Comisión Europea*.

A mediados de abril ante el tribunal de la Unión empezó el caso-maratón, durante el cual el consorcio estadounidense *Microsoft* lucha por la anulación de las penas por el presunto uso abusivo de su posición en el mercado. Éste puede ser el proceso más importante contra un monopolio en la historia de la Unión Europea. La empresa informática más grande del mundo solicita que el Tribunal de la Primera Instancia anule la decisión de Bruselas de hace dos años.

El 24 de marzo de 2004 la Comisión Europea juzgó que *Microsoft* emplea de manera excesiva la posición dominante en el mercado y puso sobre el consorcio una multa récord de 497 millones de euros. La Comisión ordenó también que el consorcio cambiara su estrategia de negocios y empezara a vender su sistema operativo Windows también en la versión privada del reproductor de multimedia *Windows Media Player* (WMP) y comparta con la competencia (después de pagar) una parte del conocimiento sobre el funcionamiento del sistema operativo Windows.

Justamente estas dos reclamaciones son las que más tocaron *Microsoft*. Pagar incluso una multa tan grande no sería problema para tal empresa cuyos ingresos netos del año pasado llegaron a los 12,25 mil millones de dólares. Sin embargo, la estrategia cuestionada por Bruselas es la base de la expansión de *Microsoft*. Aprovechando el hecho de que en un 90 por ciento de los ordenadores que funcionan en el mundo se encuentra instalado el sistema Windows, *Microsoft* poco a poco debilita a los competidores en otros segmentos del mercado.

Al vender el sistema operativo junto con WMP, *Microsoft* minó a los productos competidores (por ejemplo *RealPlayer* de la empresa *RealNetworks*) – ya que los consumidores no sintieron la necesidad de instalar aplicaciones competidoras. En cambio, las empresas que fabrican las aplicaciones para soportar los denominados servidores también perdieron ya que Microsoft no comparte información sobre cómo se comunican los servidores con el sistema Windows. Por ello las aplicaciones de los competidores ofrecen menos rendimiento.

El consorcio de Redmond rechaza estos reproches y trata de convencer de que la decisión de la Comisión Europea es radicalmente injusta ya que bloquea la innovación. – El precio del proceso es el conocimiento de si las empresas pueden mejorar sus productos por medio de añadir a ellos las nuevas posibilidades y si la empresa que tenga éxito debe compartir sus conocimientos con la competencia – dice el mensaje oficial de Microsoft publicado antes del juicio de lunes.

Será un espectáculo en sí mismo. Las partes serán escuchadas no por cinco jueces (como siempre) sino que hasta 13 integrarán la denominada *Gran Cámara del Tribubal de Primera Instancia* presidida por *Bo Vesterdorf*, juez – presidente de SPI. Ambas partes serán representadas por los mejores abogados. El presidente de Microsoft Steve Ballmer presenciará el juicio personalmente.

La Comisión Europea está convencida de que su análisis legal del año 2004 se defenderá ante el tribunal. Tenemos buenos argumentos – aseguró el viernes el comisario de competencia *Neelie Kroes*. Los empleados de la Unión Europea están soportados por la coalición de tales empresas como *IBM, Nokia, Oracle* y *Sun Microsystems*.

La importancia del juicio crece ya que en un momento toda la historia puede repetirse. A principios del año que viene Microsoft piensa publicar en el mercado el nuevo sistema operativo – Vista. Le acompañarán las nuevas aplicaciones: cortafuegos (competitivas incluso ante, por ejemplo, los productos de *Norton* o bien *McAfee*), creación y lectura de documentos electrónicos (competitivas ante *Adobe Acrobat*) etc.

Virus de plataforma múltiple

En Internet apareció un virus que infecta tanto el sistema *Windows* como *Linux*.

El virus fue encontrado por los empleados de Kaspersky Lab; se le asignó el nombre doble: *Virus.Linux.Bi.a/Virus.Win32.Bi.a*.

El código de prototipo (ya que no es virus sensu stricte, más bien *proof of concept*) no hace ningún daño y su funcionamiento está muy limitado. No es capaz de moverse por su propia cuenta, infecta los archivos solamente del directorio en el que fue instalado y para infectar es necesario que el usuario descargue el virus de Internet y lo inicie.

El peligro de infectarse es mínimo, sobre todo en el caso de los ordenadores con el sistema Linux, ya que las personas que trabajan con él pocas veces emplean los permisos del administrador y cuando ya ejecuten las aplicaciones externas será justamente en los directorios creados especialmente para ellas.

Además, *Linux.Bi.a/Virus.Win32.Bi.a* no es la primera prueba de que sea posible escribir un virus que funcione en dos plataformas.

Probablemente fue creado por una persona de la *antigua escuela* de escribir virus que demostró así que era capaz de escribir tal código. La mayoría de las aplicaciones nocivas que se escriben actualmente se hacen a demanda de grupos especializados para los cuales infectar un ordenador es una forma de ganar dinero. En el interior del virus se encontraron unas palabras y lemas, entre otros: Greetz to: *Immortal Riot, #RuxCon!. RuxCon* que es popularizado en los años noventa e-zin electrónico creado por especialistas de seguridad.



Figura 1. *Sitio web de Kaspersky Lab*

Los trabajos sobre el código de *Linux.Bi.a/Virus.Win32.Bi.a* se realizan todo el tiempo en forma que se parece al ciclo de creación de las populares aplicaciones Open Source. Gracias a ello, las pruebas realizadas demostraron que el virus no quería trabajar bajo las versiones más nuevas del kernel de la serie 2.6.

Después de conocer mejor la serie 2.4 salió a la luz de que la culpable es probablemente la mezcla del empleo manual en el ensamblador de la antigua llamada del sistema, pequeño error durante el proceso de registros por GCC y del cambio de la configuración estándar de los kernels 2.6.16.x. ¡El autor del parche que le deja al virus el funcionamiento en los kernels más recientes es ... el mismo Linus Torvalds!

El empleo en el código del virus de la abandonada ya llamada del sistema puede sugerir que *Virus.Linux.Bi.a* fue creado por el fabricante de aplicaciones antivirus solamente para objetivos de publicidad. Tales experimentos empezarán a tener importancia real cunado en los sistema Linux se popularicen los sistemas de instalación y ejecución de las aplicaciones por lo usuarios a su propia cuenta, por ejemplo *Klik* ó *FUSE*.

Los especialistas de Kaspersky Lab subrayan que el código Bi.a puede emplearse para crear aplicaciones mucho más maliciosas. Además, creen que en el futuro aparecerán muchos más virus que serán capaces de infectar al mismo tiempo *Windows, Linux* y *Mac OS X*.

Google compra nuevo algoritmo

El estudiante israelí *Ori Alon* vendió a la empresa Google los derechos al nuevo algoritmo de búsqueda de texto en páginas Web. La tecnología llamada Orion forma parte de su tesina doctoral, de la cual trabaja en la *Universidad de la Nueva Gales del Sur*.

El creador del nuevo algoritmo dijo que sigue trabajando sobre él y la versión final debe estar preparada durante los 18 meses. Las fuentes próximas a la empresa Google informaron que el israelí abandonó ya su universidad y es empleado de la central del consorcio donde continuará los trabajos sobre la nueva tecnología.

Las autoridades de la *Universidad de la Nueva Gales del Sur* informaron que se llevaban las conversaciones sobre la compra del algoritmo Orion también con las empresas Microsoft y Yahoo.

Actualización de Bagle

Durante los últimos días los ordenadores infectados por el gusano Bagle empezaron a descargar actualizaciones, dotando al insecto de unas funciones nuevas – informan los representantes de la empresa *F-Secure*. De sus análisis resulta que el objetivo principal de la actualización es instalar una nueva herramienta, de mucho más rendimiento para enviar spam.

De la información presentada por *Mikko Hypponen*, jefe del departamento de investigación de *F-Secure*, resulta que los autores de Bagle empezaron una amplia operación de actualización del gusano. No se sabe exactamente cuántos ordenadores en el mundo se quedan infectados por este insecto, sin embargo, según *F-Secure* en todo el mundo las actualizaciones se instalaron en miles de máquinas.

Las empresas que se ocupan de la fabricación de las aplicaciones antivirus tratan de parar el proceso de actualización – al bloquear los servidores que comparten update. Desgraciadamente, los autores de Bagle no están ociosos – pronto, después de desactivar un servidor la actualización aparece en el siguiente (por el momento se cerraron ya los servidores hospedados en Eslovaquia y en EE.UU).

Contenido del CD

En el disco que acompaña a la revista se encuentra *hakin9.live* (*h9l)* en la versión 3.0 - aur – distribución bootable de Aurox que incluye útiles herramientas, documentación, tutoriales y material adicional de los artículos. Para empezar el trabajo con *hakin9.live*, es suficiente ejecutar el ordenador desde el CD. Después de ejecutar el sistema podemos registrarnos como usuario hakin9 sin introducir contraseña.

El material adicional se encuentra en los siguientes directorios:

- *docs* – documentación en formato HTML,
- *hit* – titulares del número: Safety Lab Shadow Database Scanner – el mejor scaner para las bases de datos, Outpost PRO Firewall 3.51,
- *art* – material complementario a los artículos: scripts, aplicaciones, programas necesarios,
- *tut* – tutoriales, tutoriales tipo SWF,
- *add* – libros y documentación en formato PDF: *Linux IPv6 HOWTO, Securing Debian Manual, Snort Users Manual, SQL Injection Protection,*
- *rfc* – conjunto de documentos RFC actuales.

Los materiales antiguos se encuentran en los subdirectorios _arch, en cambio, los nuevos – en los directorios principales según la estructura mencionada. En caso de explorar el disco desde el nivel de arranque de *hakin9.live*, esta estructura está accesible desde el subdirectorio */mnt/cdrom*.

Construimos la versión 3.0 – aur *h9l* en base a la distribución de Aurox y de los scripts de generación automatica (*www.aurox.org/pl/live*). Las herramientas no accesibles desde el CD se instalan desde el repositorio de Arox con el programa *yum*.

¡Atención!

Safety Lab ofrece a los lectores de hakin9 la versión completa de Shadow Database Scanners para 2 direcciones IP y con una duración de 30 días.

Al enviar un correo electrónico a *support@safety-lab.com*, debes introducir 2 direcciones IP del Servidor de las Bases de Datos.

La versión completa será válida durante los 30 días a partir de la fecha de recepción de las direcciones IP. Para recibirla, por favor, ponte en contacto a *support@safety-lab.com* escribiendo en en asunto del email *hakin9-safety-lab-offer.* Válida hasta el 30 de Septiembre de 2006.

Safety-Lab

http://www.safety-lab.com/en

En comparación con la versión 2.9.1-ng *h9l*, el cambio mas importante es el sistema basado en distribución de Aurox Live 11.1 y el traspaso de Flubox al entorno gráfico KDE.

Tutoriales y documentación

La documentación está compuesta de, entre otros, tutoriales preparados por la redacción que incluyen ejercicios prácticos de los artículos *Problemas con atenticación HTTP, Análisis del tráfico en la Red*

Suponemos que el usuario emplea *hakin9.live*. Gracias a ello evitaremos los problemas relacionados con las diferentes versiones de los compiladores, la diferente localización de los archivos de configuración u opciones necesarias para ejecutar la aplicación en el entorno dado. ●

Figura 1. *Booteando hakin9 live*



Figura 2. *Pantalla de bienvenida*

Si no puedes leer el contenido del CD y no es culpa de un daño mecánico, contrólalo en por lo menos dos impulsiones de CD.



En caso de cualquier problema con CD rogamos escriban a: *cd@software.com.pl*



TTpU – TDFS's TCP/IP Packets Unlimited

Sistema Operativo: Linux
Licencia: distribución libre, descargar y usar.
Aplicación: generador de paquetes TCP/IP
Página Web: *http://www.poetidistrada.com/ttpu/*
Autor: Alberto Maria Scattolo

TTpU es una herramienta escrita para que pueda generar cualquier tipo de paquete TCP/IP con la posibilidad de especificar muchas de las opciones de IP y TCP.

La mayoría de la conexiones en una red se basan en TCP e IP. El TCP define como establecer y cerrar una conexión, como mantener los paquetes en orden y como comportarse en el caso de que falte algún paquete o haya algún error. El TCP hace esto usando opciones especiales, las TCP flags y los números de secuencia y de reconocimiento. Esas operaciones son dirigidas por el sistema operativo para que los usuarios no tengan que preocuparse por ellas, pero, ¿Qué pasaría si pudiéramos especificar la IP de origen y la de destino, el puerto de origen y de destino, la secuencia y los números de reconocimiento, las TCP flags, el tamaño de la ventana y datos opcionales para que fueran mandados con un paquete? TTpU lo hace. Las posibilidades son muchísimas. Para ver el tráfico TCP es necesario un sniffer trabajando en modo promiscuo.

Con TTpU podemos hacer muchos tipos de escaneos, por ejemplo, escaneos de *syn* y de *conectar.*

El concepto de escaneo *syn* es muy simple – mandamos un paquete con una flag SYN al host remoto a un puerto determinado. Si el puerto está cerrado deberíamos recibir un paquete con flags RST y ACK. Si en cambio no recibimos nada podemos sospechar que hay algo (un cortafuegos) que está bloqueando ese paquete, así que podemos asumir que el puerto está filtrado.

Si queremos hacer el escaneo de conectar, tenemos que intentar establecer una conexión mandando un paquete SYN. Si el puerto remoto está abierto recibiremos un paquete SYN y ACK enviado por el host remoto.

Ésta es la sintaxis de TTpU:

```
# ttpu <network interface> <source IP>
 <source port> <destination ip>
 <destination port> <sequence number>
 <acknowledgment number> <urg> <ack>
 <psh> <rst> <syn> <fin> <window size> [data]
```

Asumiendo que quisiéramos escanear el puerto 80 del host 192.168.0.94, el código sería:

```
# ttpu eth0 192.168.0.23 32456 192.168.0.94 80 3718131341 0
                       0 0 0 0 1 0 8192
```

Para poder inyectar datos en una conexión abierta, tenemos que saber algunas cosas fundamentales. Cada paquete TCP almacena dos números, el de secuencia y el de reconocimiento. Estos números son usados para mantener los paquetes en orden de manera que el sistema pueda determinar como procesarlos.

Por ejemplo, asumiendo que el servidor objetivo fuera 192.168.0.94:65534, usando un sniffer sabríamos que el cliente origen está en 192.168.0.23:33128, el número de secuencia es 1674837801 y el número de reconocimiento 1682618503. Si mandamos un paquete con los números de secuencia y reconocimiento correctos, el servidor lo aceptará como válido, pero si lo marcamos con flags sin sentido tales como SYN PSH ACK, el servidor nos devolverá un paquete RST y la conexión se cerrará:

```
# ttpu eth0 192.168.0.94 33128 192.168.0.23 65534 1674837801
                       1682618503 0 1 1 0 1 0 8192
```

Con TTuP podemos llevar acabo algunas acciones para comprobar las vulnerabilidades de un sistema operativo remoto.

Se sabe que algunos sistemas operativos son vulnerables a paquetes que declaren un mismo host y puerto de origen y de destino. Windows XP SP2 (con el cortafuegos apagado) es uno de ellos. Asumiendo que 192.168.0.94 estuviera corriendo un sistema vulnerable a este tipo de ataque y que el puerto 139 estuviera abierto.

```
# ttpu eth0 192.168.0.94 139 192.168.0.94 139 1674837801 0 0
                       1 1 0 1 0 8192
```

En un sistema Windows XP SP2 esto resultaría en 15 segundos de denegación de servicio y un usó de la CPU del 100%.

Alberto Maria Scattolo
thedarkfreesoul@poetidistrada.com

Tema caliente

Network Defense

Victor Oppleman 

Grado de dificultad

Una técnica de seguridad poco conocida que ha demostrado ser uno de los medios de defensa más efectivos contra los ataques de denegación-de-servicio (denial-of-service).

Se ha utilizado globalmente por los proveedores de servicios de internet (ISP) como una manera de proteger a sus receptores. Como se explicará en este artículo la técnica, conocida como *sinkholing*, también puede usarse para proporcionar valiosa información con respecto a las amenazas a las que se enfrenta tu red. Con el empleo de la técnica de sinkhole (o sumidero) ganarás otros medios para defender tu red y obtener información valiosa sobre las amenazas y las configuraciones erróneas significativas que haya en esta.

Este artículo, escrito para usuarios conocedores de la red, os proporcionará lo siguiente:

- Los antecedentes y funcionamiento de la Sinkhole – breve reseña de los sinkholes en las IP y cómo varias organizaciones lo han puesto en práctica con éxito.
- La utilización de señuelos de red – cómo pueden usarse las técnicas de sinkhole aplicadas con el empleo de darknets y honeynets para atrapar y analizar los sondeos malintencionados, los intentos de infiltración, y otros eventos, junto a elementos de monitorización de la red, como la detección de intrusiones.
- La protección contra la denegación de Servicio – cómo las organizaciones y sus ascendentes proveedores de servicios de Internet han creado un medio de protección contra la denegación de servicio a través de la utilización extensiva, y puntual, de la técnica de sinkhole (o de sumidero).
- La dispersión inicial (Backscatter) y el rastreo de origen (Tracebacks) – explicación breve acerca de la dispersión inicial y cómo pueden usarse los rastreos de origen para identificar el punto de ingreso de un ataque

En este artículo aprenderás...

- Aprenderás a usar las técnicas del sinkholing y cómo protegerse de los ataques de denegación de servicio.

Lo que deberías saber...

- Debes tener conocimientos básicos sobre los ataques de denegación de Servicio
- Debes conocer los problemas del tráfico de red desde los ISP (Proveedores de Servicios de Internet)

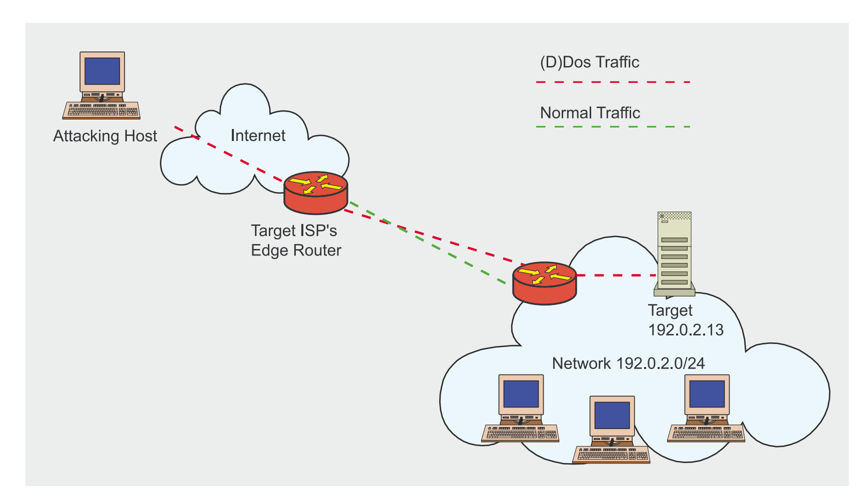


Figura 1. *Ataque a la dirección IP 192.0.2.13 (antes del sinkholing)*

de denegación-de-servicio en una red de gran tamaño.

Antecedentes y Funcionamiento

En este texto, el término *sinkhole* puede definirse como un medio generalizado de redireccionar el tráfico de red de un IP específico debido a diferentes razones relacionadas con la seguridad, dentro de las que se incluyen el análisis forense, la diversificación de ataques y la detección de actividades anómalas. Tier-1 ISPs fueron los primeros en poner en práctica estas tácticas, generalmente para proteger a sus receptores. Desde entonces, se han adaptado las técnicas para recoger información de interés relacionada con las amenazas en la red, a fin de realizar análisis de seguridad. Para visualizar la forma más simple de sinkhole, ten en cuenta lo siguiente:

Un tráfico malintencionado y perjudicial proveniente de diversas redes tiene como destino la red 192.0.2.13, como se muestra en la Figura 1. La organización que es objeto o blanco de este tráfico utiliza la 192.0.2.0/24 como su bloque de dirección de red que es enrutada por su flujo ISP ascendente. El ataque que se hace débil interrumpe las operaciones de negocio de la organización blanco e incrementa potencialmente sus costos debido al aumento del uso de la amplitud de banda y la necesidad de acción por parte del ISP debido a que la cantidad abrumadora de tráfico generado por el ataque perjudica a los clientes adyacentes, como una forma de daño colateral.

El ISP reacciona e inicia temporalmente un tipo de sinkhole en forma de agujero negro inyectando una ruta más específica para el objeto o blanco (192.0.2.13/32) dentro de su segmento principal, cuyo próximo salto es la interfaz de desechos en su router edge (también conocido como el *null0* o el *bit bucket*), como se muestra en la en Figura 2.

Esta táctica redirecciona el tráfico ofensivo hacia el sinkhole del ISP en lugar de permitirle llegar al blanco u objeto original. Lo beneficioso se manifiesta desde el mismo momento en que el sinkhole surte efecto, es muy probable que los clientes adyacentes del ISP (siempre que el ISP diseñe cuidadosamente sus defensas de sinkhole) queden libres de daños colaterales y que el blanco del ataque haya recuperado el uso de su conexión de Internet y el acceso local al dispositivo específico que fue objeto del ataque. Desgraciadamente, las direcciones de IP específicas (el dispositivo) que están siendo atacadas no podrán interactuar por Internet con los sistemas remotos hasta tanto no se elimine el sinkhole (probablemente después de que el ataque haya menguado). Obviamente, los servicios originalmente proporcionados por el dispositivo que ha sido atacado pueden ser desplazados a un dispositivo alternativo con una dirección IP diferente, pero habría que tener en cuenta otras consideraciones con respecto a la caducidad del DNS TTL, y así sucesivamente.

Este ejemplo refleja sólo uno de los tipos de sinkhole, conocido normalmente como ISP-induced blackhole route (ruta de agujero negro inducida por el ISP), no obstante, el mismo debería familiarizarte con el concepto para que así podamos explicar los diferentes usos del *sinkhole*.

La utilización de Sinkholes para Desplegar Redes de Señuelo

La utilización de varios tipos de redes de señuelo con el propósito

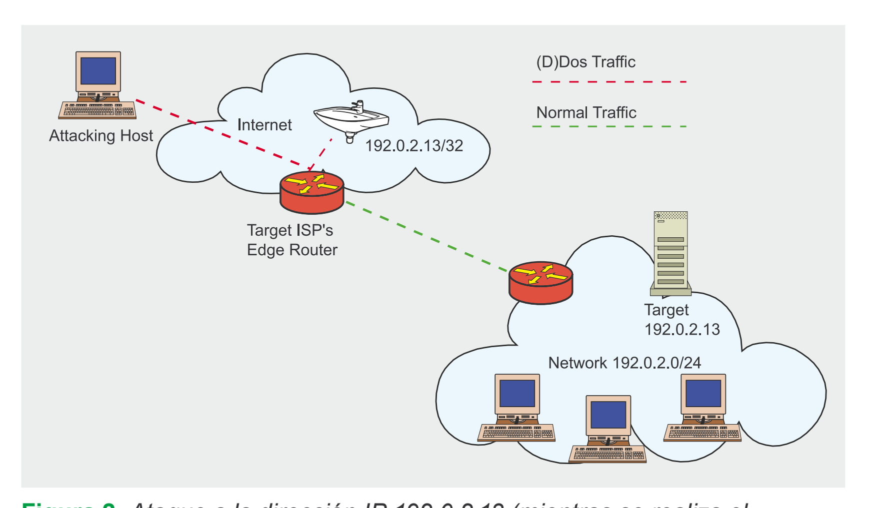


Figura 2. *Ataque a la dirección IP 192.0.2.13 (mientras se realiza el sinkhole)*



Listado 1. *Un ejemplo de la configuración del BGB*

```
router bgp XXX
redistribute static route-map static-to-bgp
# Route-map is a policy mechanism to
# allow modification of prefix attributes, or special
# filtering policies
route-map static-to-bgp permit 10
match tag 199
set ip next-hop 192.0.2.1
set local-preference 50
set community no-export
set origin igp
```

Listado 2. *La configuración básica por parte del ISP*

```
router bgp XXX
# Route-map is simply a policy mechanism
# to massage routing information such
# as setting the next hop
neighbor < customer-ip >
 route-map customer-in in
# prefix-list is a static list of customer
#prefixes and mask length that
# are allowed.
#Customer should be allowed to
# announce down to a single host
# in their prefix(es) such as 172.16.0.1/32
neighbor < customer-ip > prefix-list 10 in
# ebgp-multihop is necessary to prevent
# continuous prefix announcement and
# withdrawal
neighbor < customer-ip > ebgp-multihop 2
# Now we define the
#route-map for policy match
# and setting the blackhole
# next hop
route-map in-customer permit 5
# the customer sets
#this community on their side,
# and the ISP matches on its
# side.  XXXX  would likely be
#the customer ASN,
# and NNNN is an arbitrary number agreed
# on by the ISP and the customer
match ip community XXXX:NNNN
set ip next-hop < blackhole-ip>
set community additive no-export
```

de entrampar, exponer, y acopiar información valiosa constituye una forma más novedosa de emplear los sinkholes.

Un señuelo es algo que conduce a una trampa, un cebo que engaña y conduce al peligro, al poder del enemigo, actuando como carnada.

Los dos tipos de redes de señuelo que analizaremos en detalle son la darknet (red oscura) y la honeynet (red de miel). Ambas son útiles para recopilar información valiosa relacionada con asuntos de seguridad, pero una de ellas es particularmente útil en el campo de la ingeniería de redes seguras.

El Despliegue de las Darknets

En general, una darknet (red oscura) es una porción de un espacio IP asignado y enrutado donde no reside ningún servicio sensible. Tales redes se clasifican de *oscuras* porque aparentemente no hay nada *encendido* en ellas. Sin embargo, una red oscura (darknet) de hecho incluye por lo menos un servidor, diseñado para actuar como un vacío de paquete. Este servidor recoge y organiza los paquetes que entran en la red oscura, y son útiles para los análisis a tiempo real o para el análisis forense de la red posterior al evento.

Cualquier paquete que entra en una red oscura es inesperado. Puesto que ningún paquete legítimo debe aparecer dentro de una red oscura, aquellos que llegan lo hacen debido a algún error de configuración, o por la causa más frecuente, que es que haya sido enviado por software maligno. Este software maligno, en su búsqueda de dispositivos vulnerables, enviará paquetes hacia el interior de la red oscura, y por tanto, se expondrá a las revisiones de seguridad administrativas. Hay un enfoque ingenioso en este método para encontrar gusanos y otros softwares malignos de propagación. Sin falsos positivos, y sin firmas o instrumentos complicados de análisis estadísticos, un administrador de seguridad con redes oscuras empleadas correctamente puede detectar las exploraciones (intentos llevados a cabo por softwares malignos con el fin de descubrir receptores adyacentes convenientes para la propagación) en redes de cualquier tamaño. Ésa es una herramienta de seguridad poderosa. Además, los paquetes que llegan a la darknet revelan configuraciones erróneas de redes que son inofensivas y cuya eliminación agradecerán los administradores de redes. Por supuesto, las darknets tienen usos múltiples en el terreno de la seguridad. Pueden usarse para albergar a los recaudadores de flujo, a los detectores de backscatter (o dispersión inicial), a los rastreadores de paquetes y a los sistemas de detección de intrusión. La elegancia de la darknet o red oscura es que reduce considerablemente los positivos falsos de cualquier dispositivo o tecnología mediante una simple reducción del tráfico.

La puesta en funcionamiento de un darknet o red oscura es relati-

¿Quieres recibir tu revista regularmente?

¿Quieres pagar menos?

¡Pide suscripción!



hakin9

por suscripción es más barata: 38 €

Ahora ¡un regalo! para los suscriptores: 6 números anteriores de hakin9 en versión electronica

Pedido

Por favor, rellena este cupón y mándalo por fax: 0048 22 887 10 11 o por correo: Software-Wydawnictwo Sp. z o. o., Piaskowa 3, 01-067 Varsovia, Polonia; e-mail: *subscription@software.com.pl*

Para conocer todos los productos de Software-Wydawnictwo Sp. z o. o. visita *www.shop.software.com.pl*

Nombre(s) .... **Apellido(s)** ....

Dirección ....

C. P. .... **Población, provincia** ....

Teléfono .... **Fax** ....

E-mail .... **Suscripción a partir del N°** ....

Precio de suscripción anual de hakin9: 38 €

Realizo el pago con:

☐ **tarjeta de crédito** (EuroCard/MasterCard/Visa/American Express) **nº** ⎿⏌⏌⏌⏌⎿⏌⏌⏌⏌⎿⏌⏌⏌⏌⎿⏌⏌⏌⏌ **CVC Code** ⎿⏌⏌⏌

Válida hasta ⎿⏌⏌⏌⏌

☐ **transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO**

Número de la cuenta bancaria: 0049-1555-11-221-0160876

IBAN: ES33 0049 1555 1122 1016 0876

código SWIFT del banco (BIC): BSCHESMM

Fecha y firma obligatorias:

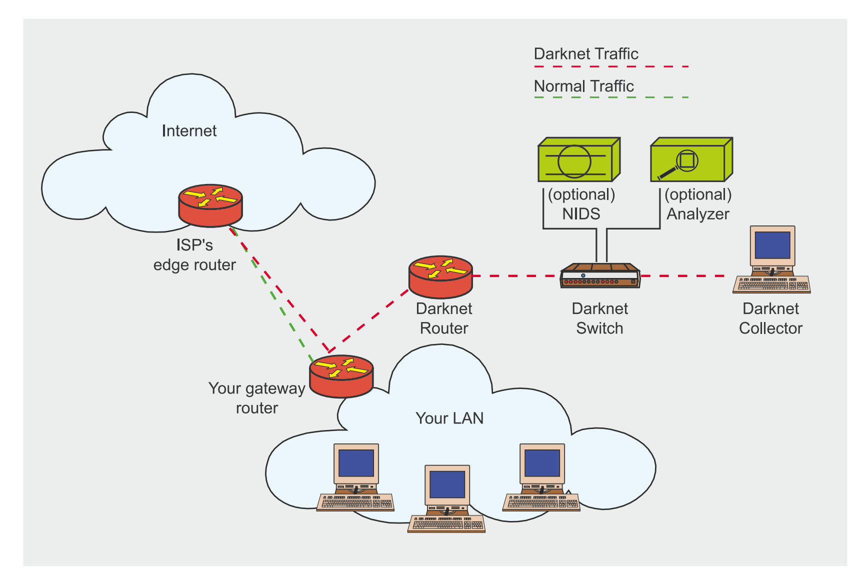


Figura 3. *Topología física de referencia para las redes oscuras o darknets*

vamente sencilla. De hecho, aquí tienes cinco pasos fáciles.

Seleccione una o más regiones inutilizadas en los espacios de las direcciones IP de su red, que enrutarás con su red oscura o darknet. Esta podría ser un prefijo /16 o mayor de las direcciones, o por lo contrario reducirse a una dirección única (/32). Una mayor cantidad de direcciones propiciará una percepción más exacta desde el punto de vista estadístico de la actividad de red no solicitada. Yo recomiendo seleccionar, por ejemplo, varios segmentos de direcciones, tales como un /29 de cada una de las diferentes redes internas, y un /25 de su asignación de red pública (externa). No existe razón alguna por la que no puedas hacer una darknet en una región de su espacio interno de dirección privada (por ejemplo, espacio RFC 1918, 10.0.0.0/8). De hecho, si seleccionas regiones de tu red interna para convertirlas en darknet o redes oscuras, podrás observar sondeos o exploraciones internas que podrían dejarse de ver si solamente creas redes oscuras en los segmentos de redes externas (públicas). Otra estrategia que puede tenerse en cuenta en las organizaciones que utilizan un enrutado específico para sus redes internas, es apoyarse en la regla de enrutamiento que plantea que *la ruta más específica es la vencedora* (usualmente se distribuye a través de algún tipo de protocolo de pasarela interior). Esto quiere decir que si utilizo las redes 10.1.1.0/24 y 10.2.1.0/24 internamente, puedo enrutar sencillamente la red 10.0.0.0/8 por completo hacia mi darknet. Entonces sé que si mi red está configurada adecuadamente, la darknet recibirá todo el tráfico de la 10.0.0.0/8, salvo las redes dentro de esta que estoy enrutando/utilizando de manera específica (las cuales probablemente tienen entradas de enrutamiento estáticas en mi infraestructura de red).

El próximo paso sería la configuración de la topología física. Necesitarás un enrutador o conmutador (layer-3) que remitirá el tráfico a tu darknet, un servidor con una amplia capacidad de almacenamiento para servir como recolector de datos, y un conmutador de Ethernet que usarás en el futuro para conectar estos componentes y los componentes opcionales, tales como un sensor de IDS o un analizador de protocolo. Como enrutador puedes elegir el uso de un dispositivo de pasarela existente, ya sea interno o externo (o ambos, aunque no es recomendable) – La mayoría de las redes oscuras de las empresas, a diferencia de las darknets de los soportes de telecom, se localizan dentro de uno de los DMZs de la organización y se encuentran separadas del resto de la red. Por consiguiente, podrías tener en cuenta para realizar este trabajo el uso de un firewall en lugar de uno de tus enrutadores. No obstante, nuestra recomendación es que utilices su enrutador de pasarela externo para las darknets o redes oscuras externas, y un conmutador interno layer-3 para tus darknets internas. En cualquier caso, lo más importante a tener en cuenta es que configurarás este dispositivo de enrutamiento para remitir el tráfico destinado a la darknet que este recibe desde fuera de una interfaz ethernet dedicada a la darknet (a través del conmutador), hacia el servidor recolector que configurarás para que acepte dichos paquetes. El servidor recolector también debe tener una interfaz dedicada a la darknet que recibirá esos paquetes. Para la administración, el servidor recolector también necesitará por lo menos una interfaz de Ethernet adicional (que será ubicada en una LAN de administración separada). Asegúrese de emplear sus propias y mejores prácticas para la seguridad de los dispositivos de red, pues le garantizamos que muy pronto todo tipo de cosas desagradables fluirán por este segmento de red. Controla el impulso de utilizar rápidamente un conmutador DMZ existente con el propósito de conectar estos componentes, a menos que puedas configurar adecuadamente la VLAN de manera que ningún paquete de transmisión logre llegar a la red oscura o darknet. —recuerda que la darknet es sólo para el trafico ilegítimo, por tanto no es conveniente que las trasmisiones correctas provenientes de tus otras LAN invadan el territorio de la red oscura. La Figura 3 ilustra un ejemplo de esta configuración. En nuestros ejemplos empleamos un enrutador o conmutador ejecutando el Cisco IOS con una licencia de programa layer-3, un servidor FreeBSD-based, y un interruptor layer-2 no administrado para conectar los dispositivos.

A fin de que nuestro servidor recolector evite el protocolo de resolución de direcciones (ARP) para cada dirección en el espacio de la darknet, configuraremos el enrutador para

que remita el tráfico destinado a la darknet hacia una única dirección IP final en la interfaz de Ethernet del servidor para la red oscura. Para lograr esto, sugerimos dedicar un /30 de red como el 192.0.2.0/30 para el recorrido entre el enrutador y la interfaz de la darknet. Esto haría que la interfaz Ethernet de su enrutador fuese la 192.0.2.1/30 y podría llegarse al servidor recolector vía la 192.0.2.2/30. La configuración de la interfaz depende en gran medida de las plataformas que has seleccionado, por tanto nosotros asumiremos que estás en condiciones de hacerlo por tu cuenta. En nuestros ejemplos, estamos usando el Cisco IOS con la licencia de software del layer-3. Una vez se haya realizado esto, simplemente introducirás las declaraciones de enrutamiento adecuadas en el conmutador para que remita todo el tráfico de su red oscura o darknet a la 192.0.2.2 en el servidor colector, y ya estás fuera de peligro:

```
router#conf t
router(config)# ip route 10.0.0.0 ↵
255.0.0.0 192.0.2.2
router(config)# ^Z
router# wr
```

Ya debes estar recibiendo el tráfico de la red oscura o darknet. Un ejemplo de la topología lógica se muestra en la Figura 4.

Qué hacer con el tráfico una vez que este llegue allí es otra historia. El servidor debe configurarse para que no responda a ningún dato que reciba en su interfaz de red oscura. Por supuesto, llevará a cabo un protocolo de resolución de direcciones (ARP) para tu dirección configurada (solamente la 192.0.2.2/30) con el fin de establecer comunicación con el enrutador, no obstante, todos los otros paquetes deben ser desechados por algún tipo de firewall en el dispositivo local. Como decía con anterioridad, no debe existir ningún tipo de administración en la interfaz de la darknet, deberás configurar otra interfaz de Ethernet en la que llevarás a cabo las funciones de administración y gestión. La ruta predeterminada para el sistema debe ser la pasarela de la interfaz de administración. En cuanto al firewall necesario, tu selección de plataforma del servidor tendrá que ver con la selección del firewall, pero recomendamos que emplee un sistema basado en BSD y en pf, o un ipfw2 como firewall. Si debe habilitarse o no un loggin para su firewall dependerá en gran medida del uso que le vayas a dar. Nosotros utilizamos herramientas de análisis de archivos de registro que exigen un loggin para activarse (de manera que los registros puedan ser analizados sintácticamente y se generen las alertas) Sin embargo, en dependencia de las diferentes elecciones de software y hardware, y del tamaño de su red oscura o darknet, este logging puede degradar seriamente la eficacia de la darknet. Como medida de seguridad adicional (los firewalls pueden dañarse o apagarse accidentalmente) sería buena idea anular la ruta del tráfico de la darknet en caso de que accidentalmente este no sea filtrado. Un ejemplo de anulación de ruta según FreeBSD podría ser este:

```
route add -net 10.0.0.0/8 ↵
127.0.0.1-blackhole
```

Ahora que tu darknet está funcionando y ya has protegido su servidor recolector de la darknet, necesitas guardar los datos en un formato útil para el análisis y las herramientas forenses. La opción más obvia serían los archivos binarios formateados para pcap (capturar paquetes) pues son prácticamente ubicuos y la mayoría de las aplicaciones de análisis de red pueden operar en ellos. La manera más fácil de hacerlo de forma continuada es mediante el uso de la opción de rotación incorporada del programa tcddump. El programa tcpdump es proporcionado por el Grupo de Investigación de Red del Lawrence Berkeley National laboratory. Según nuestra opinión el siguiente es un ejemplo de la fórmula del comando del tcddump para lograr la rotación del registro:

```
tcpdump -i en0 -n -w darknet_dump -C125
```

En este ejemplo, se le ordena al tcpdump escuchar en la interfaz en0, la resolución (DNS) número-a-nombre se desactiva, y un archivo nombrado darknet_dumpN se escribe por cada 125 millones de bytes asignados, donde N se incrementa para que los nombres de los archivos sean únicos. Repetimos, esto proporcionará un archivo binario formateado para pcap que contiene el tráfico de la red. Luego podrás usar este archivo como entrada en tus softwares favoritos de análisis de red. La idea aquí es guardar una copia de los datos

Listado 3. *La configuración básica del cliente*

```
router bgp XXXX (customer's ASN)
# the customer will install a static route,
# which is redistributed into BGP
# hereredistribute static route-map
# static-to-bgp
# just like the ISP, use a
# route-map to set
# and match specific prefix
# attributes
route-map static-to-bgp permit 5
# match the arbitrary tag,
# agreed on by the customer and the ISP
match tag NNNN
set community additive
# XXX:NNNN
# NNNN is the tag, agreed on
# by the customer and the ISP
ip route 192.168.0.1 255.255.255.255
Null0 tag NNNN
```

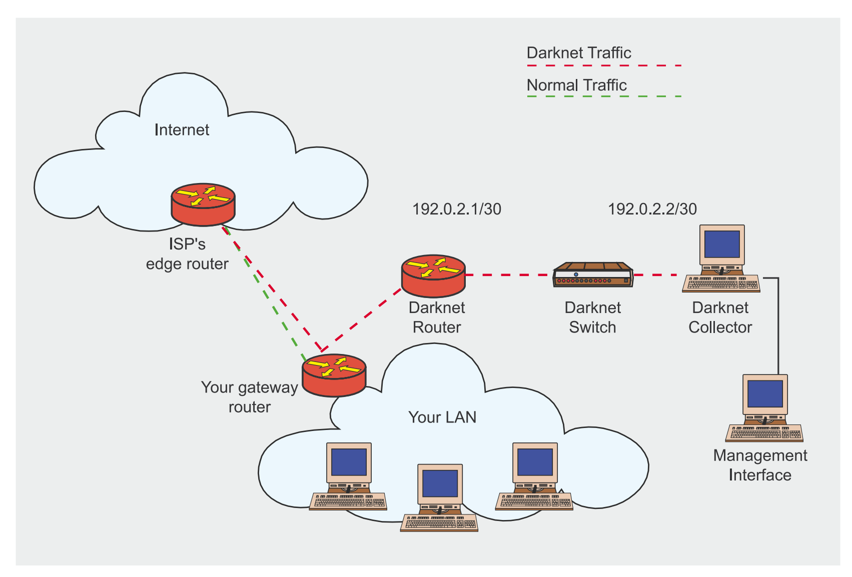


Figure 4. *Una referencia de topología lógica para las redes oscuras*

y utilizar abundantes herramientas diferentes para reproducir los archivos posteriormente, en busca de características interesantes del tráfico. En condiciones normales, emplearías un programa como el tcpdump con una expresión específica de BPF (filtro de paquete Berkeley) para buscar cosas dentro de estos archivos. A pesar de que esto puede hacerse en tiempo real (tiempo de captura), si guardas un registro de todo el trafico, podrás utilizar diferentes herramientas con posterioridad sin correr el riesgo de perder algo importante.

Otras herramientas útiles que facilitan la visualización de flujos de tráfico son el Argus, la red *Audit Record Generation* (Generación de datos de auditoria) y el Sistema de Utilización, creadas por QoSient. Aunque su configuración es demasiado compleja para explicarla aquí, nosotros utilizamos normalmente el Argus para observar flujos interesantes en nuestras redes oscuras o darknets. El Argus proporciona una aguda interfaz de resumen basada en el flujo que debe ayudarlo a entender con exactitud lo que sucede con respecto a los flujos de tráfico maligno.

Para visualizar el volumen de tráfico que entra en tu darknet, podrías apoyarte en herramientas de interfaz basadas en un contador, tales como el MRTG (vea *http://www.mrtg.org /*) de Tobias Oetiker. El MRTG puede ayudarle a crear bonitos gráficos desde un tráfico de darknet no tan bello. Existen también docenas de herramientas accesibles útiles para analizar los registros del firewall que pueden constituir alternativas rápidas y fáciles a las herramientas de análisis más complicadas como las basadas en pcap o el Argus. Ten presente los problemas de funcionamiento que afrontarás con el logging basado en el texto del filtro del paquete y subsecuentemente con el análisis de dichos archivos.

Literalmente existen docenas de herramientas que se pueden utilizar en tu red oscura. Para comenzar, esto es lo que encontrarías en alguna de las nuestras:

- Un sensor IDS (Bro, Snort, et al.)
- Un rastreador del paquete (el tcpdump descrito con anterioridad)
- Un analizador de flujo (argus, exportador de flujo de red desde el enrutador, SiLK, herramientas de flujo)
- Un analizador sintáctico de archivos de registro del firewall que puebla las bases de datos RRD para los gráficos
- El MRTG y los contadores de tráfico de gráficos
- El p0f (de Michal Zalewski) para categorizar plataformas de dispositivos infectados/contaminadores.

El Despliegue de las Honeynets

Al igual que las redes oscuras o darknets, la honeynet es en general una porción de un espacio enrutado con una IP designada. Ahora bien, en lugar de proporcionar un destino donde los paquetes van a morir, este destino imita un servicio real (o muchos servicios), y por tanto permite que ocurra la conexión (el apretón de manos) y se establezca un diálogo bidireccional completo. Una honeypot, o esta imitación del sistema de un servicio real, debe ser un recurso bien sostenido y cons-

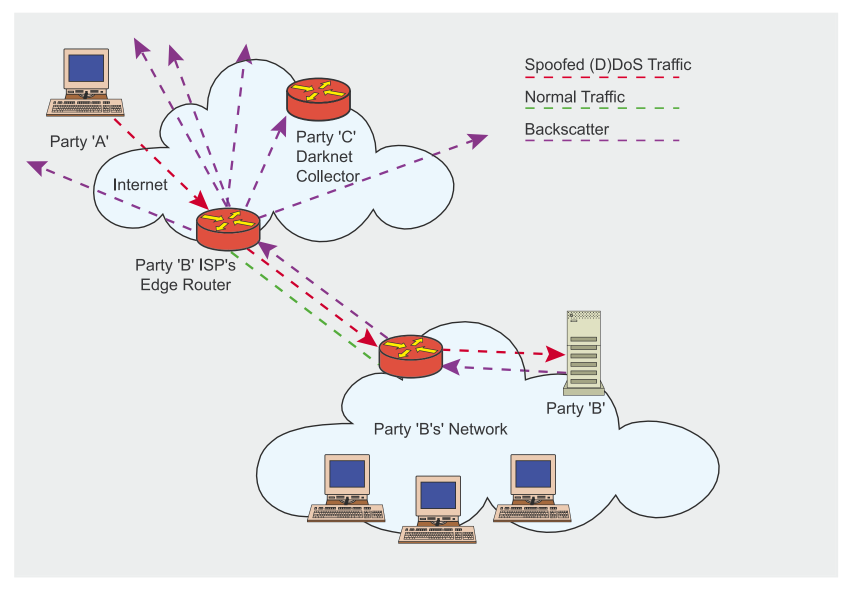


Figura 5. *Un ejemplo de backscatter durante un ataque de DDoS*

tantemente monitoreado que tenga como objetivo atraer atacantes para sondearlos y/o infiltrarlos. A pesar de que existen varios tipos de honeypots, todos persiguen la misma meta: aprender las tácticas del atacante y obtener la mayor cantidad de información posible sobre este.

Las Honeypots Físicas

Las honeypots físicas son máquinas completas dentro de la honeynet o red de miel con su propia dirección IP, con un sistema operativo y herramientas de imitación de servicios.

Las Honeypots Virtuales

Las honeypots virtuales son sistemas de programas de honeypots simulados dentro de la red de miel o honeynet que simulan condiciones de entorno tales como el sistema operativo, la pila de red, y los servicios brindados como señuelos. Un servidor físico puede proporcionar una red de miles de honeypots virtuales.

Las Honeypots de Baja Interacción

Las honeypots de interacción baja (las que más se utilizan en la actualidad) se diseñan para atraer a un atacante con una o más vulnerabilidades aparentemente explotables, establecer el diálogo, y capturar los primeros paquetes de comunicación con el atacante. Obviamente, el atacante o el software maligno autónomo que está conversando con la honeypot en algún momento se dará cuenta de que su blanco u objetivo es imposible de explotar, no obstante, antes de que eso ocurra puede quedar expuesta alguna información valiosa, léase la táctica de explotación o la firma del software maligno. Estas honeypots de baja interacción se emplean en la actualidad como modelo para las tácticas de contaminadores (spammers) (por ejemplo, el intento de derivar las heurísticas tales como las características de temporización de las transacciones SMTP de los contaminadores).

En general, existen muy pocas aplicaciones comerciales de la tecnología de la honeynet, pero la aplicación más popular se encuentra en el proyecto de fuente abierta, honeyd, de Niels Provos. Puede encontrarse más información sobre la adquisición y la instalación del honeyd en *http://www.honeyd.org*.

Datos de interés: el honeyd está diseñado para ser un honeypot/ honeynet virtual que puede simular varios sistemas operativos diferentes y componentes de software convenientes para atraer a los atacantes. Otra forma de honeypot de baja interacción que merece ser mencionada es un concepto novedoso de Tom Liston llamado LaBrea. LaBrea (llamado así por el hoyo de alquitrán) es un software demonio (servicio) que es capaz de generar respuestas autónomas a las solicitudes de conexión a través de bloques potencialmente enormes de direcciones IP. Para abreviar, crea un ambiente atractivo para el software maligno de contaminación/ propagación, pero cuenta con un truco sucio. En cuanto dicho software intenta conectarse, LaBrea retrasa, en ocasiones considerablemente, la pila de la red del remitente. Hablando en sentido figurado, la pila de la red del sistema infectado por el software maligno se queda atascada en un hoyo de alquitrán. Por tanto, no existe ninguna interacción en la capa o en el nivel de la aplicación, pero sí existe una interacción significativa en la capa 4 cuando la conexión (TCP) intenta producirse. LaBrea incluso es capaz de llevar a cabo el ARP para todas las direcciones IP virtuales que hay en su configuración sin asignarlas a las interfaces del sistema anfitrión, lo que facilita extremadamente su instalación. Puede encontrar más información sobre LaBrea en *http://labrea.sourceforge.net/labrea-info.html*.

Nota: varias agencias de investigación han llegado a la conclusión de que las honeypots de baja interacción son una táctica viable contra los gusanos de alta propagación retardándolos con el fin de proteger la infraestructura de la red. Nosotros postulamos que la configuración necesaria para lograr este beneficio es cuanto menos obtusa. No obstante, tanto LaBrea como el honeyd pueden configurarse para crear este tipo de entorno hostil para el gusano.

Las Honeypots de Interacción Alta

Las honeypots de interacción alta son menos utilizadas, pero son sumamente valiosas. El honeypot de interacción alta está diseñado para permitir que el ataque se infiltre completamente en el sistema en que reside en lugar de capturar únicamente las primeras transacciones en el diálogo entre el atacante y el honeypot. En este caso, la información que se recoge no solo incluye las técnicas de sondeo y de explotación empleadas sino que también permitirá al administrador de seguridad observar al atacante una que vez que este gane acceso al sistema y exponga de manera inconsciente sus intenciones y herramientas.

Existe una organización sin fines lucrativos conocida como The Honeynet Project (vea *http://www.honeynet.org/*) que proporciona bastante información y algunas herramientas fáciles de poner en práctica, diseñadas para permitirle al usuario la utilización de los honeypots de interacción alta. También ofrece excelentes herramientas de tipo forense para analizar los datos recogidos durante las infiltraciones en los honeypots.

Datos de interés: The Honeynet Project (*http://www.honeynet.org/*) publica varias herramientas fantásticas que puedes emplear en la utilización de tus propias honeynets. Recomendamos que prestes especial atención a las herramientas Honeywall, Termlog, y Sebek. Igualmente, el equipo del proyecto también ha escrito un libro excelente sobre la psicología, las tácticas, y las herramientas usadas por los atacantes de la manera en que se observan a través de las tecnologías de la honeynet. El libro titulado Know Your Enemy (Conoce a Tu Enemigo) que en este momento está en su segunda edición, se encuentra disponible en el sitio web de *honeynet.org*,

Tabla 1. *Los Paquetes ICMP*

Paquetes ICMP	Descripción
3.0	Red inalcanzable
3.1	Host inalcanzable
3.3	Puerto inalcanzable
3.4	Se requiere fragmentación
3.5	Fallo en la ruta de origen
3.6	Error desconocido de la red de destino
3.7	Error desconocido del host de destino
3.10	Prohibición administrativa del Host
3.11	Tipo de servicio de red inalcanzable
3.12	Tipo de servicio de host inalcanzable
3.13	Prohibición administrativa de comunicación
11.0	TTL expiró durante el tránsito
11.1	Se excedió el tiempo de desfragmentación
Paquetes de TCP	**Descripción**
RST bit set	Restablecimiento de TCP

y los beneficios que se obtienen de su venta se usan como parte de los fondos para las investigaciones del honeynet.

Recomendaciones para la utilización de las Honeynets

Para aquellas organizaciones, o aquellos que dispongan de dinero y tiempo de sobra (¿conoces a alguien?), las honeypots pueden ser una herramienta inestimable, no obstante, nosotros no recomendamos el uso de las honeypots de manera cotidiana dentro de la empresa. Sin embargo, a pesar de que no es conveniente para el uso cotidiano, cuando un software malintencionado, aparentemente inofensivo, muestra sus garras y ninguna herramienta olfateadora o forense contribuye a identificar el problema de manera que su administrador pueda resolverlo, puede utilizarse el honeynet de manera puntual con el fin de establecer la comunicación mostrándose como un blanco para este software, y por consiguiente se expondrá la información suficiente para identificar el ataque adecuadamente. Otro uso puntual que se le puede dar es como un medio para verificar una infiltración sospechosa. Por tanto, la honeynet debe ser otra flecha en la aljaba del administrador de seguridad.

Una de las puestas en práctica del honeynet que vale la pena mencionar está en uno de los principales productores de chips del mundo. Ellos tienen, en toda su red, servidores de Linux que usan VMWare, encima de los que se ejecutan cuatro máquinas virtuales, una máquina para cada una de las variedades de OS de Windows comunes en la empresa —NT, 2000, 2003, y XP. Cada una se mantiene actualizada según los niveles de ajuste estándares de la corporación. El Linux OS monitorea el tráfico y los cambios a fin de detectar gusanos nuevos (u otras amenazas) que podrían circular en la compañía. Fundamentalmente usan este entorno como una combinación de honeynet e IDS para los gusanos. Podrás encontrar más información sobre esta aplicación en *http://phoenixinfragard.net/meetings/past/200407hawrylkiw.pdf*

La puesta en práctica de los Sinkholes para defenderse contra los ataques DDoS (Blackhole Routing)

Otra utilización novedosa de la tecnología de los *Sinkholes* es como una táctica de defensa contra los ataques de denegación de servicio (distribuidos). En la sección anterior de este artículo sobre *Antecedentes y Funcionamiento*, el primer ejemplo que se dio fue la manera más sencilla que adopta esta técnica de enrutado de black-hole (agujero negro). Una vez identificado el blanco de un ataque, la dirección IP objeto del ataque era desviada hacia la interfaz de desecho hacia el límite de la red, antes de cruzar el vínculo final hacia el blanco. Esto liberaba a la red objeto o blanco de una ruptura total por saturación de vínculos, pero aún así probablemente impactaba en el funcionamiento de toda la red, especialmente de los clientes adyacentes que compartían algunas de las topologías de soporte edge con la red atacada. En la actualidad, los principales soportes de telecom han diseñado sus redes y han incluido versiones sofisticadas de este medida de defensa como parte de su filosofía de diseño para la red en general. En muchos casos, los soportes pueden utilizar una técnica de rastreo para localizar los puntos de ingreso del ataque y emplear el blakhole para los paquetes malignos allí (justo en los puntos de ingreso) en lugar de permitir que el ataque obstruya la columna vertebral del soporte y la recorra hasta llegar al vínculo de red objeto del ataque. Esta técnica de rastreo es en gran parte innecesaria puesto que las rutas de blackhole de los soportes suelen anunciarse habitualmente en toda la red entre sus enrutadores edge mediante una comunidad BGP; de ese modo, envían al blackhole el tráfico maligno en cada punto de ingreso, lo que les permite conducir los ataques al blackhole en la medida en que estos penetran y en muchos casos evitar la congestión de la columna vertebral y del edge al mismo tiempo. Algunos han ampliado incluso el control y la automatización de esta capacidad hacia el cliente final mediante lo que se conoce como blackholes a tiempo real activados por el cliente (customer-triggered real-time blackholes).

Tabla 2. *Gráfico de Resumen*

Pasos	Descripción
Comprende cómo tu ISP puede ayudarte durante un ataque de DDoS	Traza un plan de acción para enfrentar ataques de DDoS con estrategias que incrementen la capacidas de de tu IP en el área de enrutados de blackhole a tiempo real. Establece un diálogo entre tu organización y tu ISP a fin de que este le permita crear blackholes a tiempo real activados por el cliente para que se proteja, sin tener que desperdiciar tiempo valioso en sus procedimientos de escalado.
Ten en cuenta la puesta en práctica de una darknet interna	Recuerda que una darknet interna te facilita atrapar los gusanos con mayor prontitud que tu antivirus. Igualmente, esto expone configuraciones de red erróneas que te gustaría conocer
Ten en cuenta la puesta en práctica de una darknet externa	Las darknets externas pueden proporcionarte información sobre lo que está atacando tu red desde el exterior y la herramientas que se emplean pueden resultarte más fácil a la vista que las de un registro de firewall estándar. El backscatter que se recoge de una darknet externa puede proporcionarte información sobre el momento en que tu red está siendo implicada en un ataque a una tercera parte.
Explora la posibilidad de poner en práctica los honeypots con fines investigadores si dispones de tiempo y recursos.	Aunque la mayoría de las organizaciones no aprecian la implementación de una honeynet como un beneficio importante (más allá de la alerta), ésta es inestimable para los investigadores de la seguridad de la información. Considere lo que implica poner en práctica una honeynet en su organización. Al hacerlo, incluya la exploración de las leyes estatales que pueden influenciar en su determinación.

El Enrutado de Blackhole Activado

Como se ha dicho con anterioridad, la mayoría de los ISPs principales han puesto en práctica un sistema distribuido automatizado para *activar* los enrutados de blackhole en las direcciones IP que son blanco de ataques. Esta activación puede ser iniciada por el ISP o por el cliente, de forma manual o mecánica. El enrutado de blackhole activado utiliza el sinkhole sencillo que se describió con anterioridad en la sección *Antecedentes y Funcionamiento*. El sinkhole puede configurarse en todos los enrutadores de ingreso (edge) dentro de la red ISP, en la que el ISP intercambia tráfico con otros proveedores o clientes. Cuando se identifica un ataque contra un blanco de la red, el ISP o el cliente pueden anunciar el prefijo *atacado* (o un prefijo más específico) en la mesa de enrutado del BGP. El prefijo atacado se rotula con un next-hop que se enruta estáticamente hacia la interfaz de desecho de todos los enrutadores edge, y se propaga dentro de la red ISP a través de un BGP interno (iBGP). Por tanto, siempre que los paquetes destinados para el prefijo atacado penetran la red ISP (el punto del ingreso), se les envían inmediatamente a la interfaz de desecho en el enrutador más cercano anunciando el prefijo atacado.

Para que el ISP ponga en práctica el mecanismo de blackhole distribuido deben seguirse los siguientes pasos:

- Selecciona un prefijo que no esté enrutado globalmente, como el Test-Net (RFC 3330) 192.0.2.0/24, para que se utilice como next hop de cualquier prefijo atacado, que será enviado al blackhole. El uso de un prefijo de longitud 24 te permite utilizar muchas direcciones IP diferentes para tipos específicos de enrutado de blackhole. Tal vez quieras diferenciar las rutas de blackhole que utilizas para el cliente, de las internas y de las externas.
- Configura una ruta estática en cada enrutador de ingreso/escutrinio para la 192.0.2.0/24, que señale a la interfaz de desecho. Por ejemplo: ruta ip192.0.2.0 255.255.255.0 Null0
- Configure el BGP y las políticas de los mapas de ruta para que anuncien el prefijo que debe ser enviado al blackhole como se indica en el Listado 1

En la configuración del ejemplo, estamos redistribuyendo las rutas estáticas en el BGP que se corresponden o igualan al *código199* (ver más adelante), estableciendo el next hop en una dirección IP que está enrutada hacia la interfaz de desecho, estableciendo la preferencia local en 50 (el menos preferido), y asegurándonos así de que estas rutas no se infiltrarán en ninguno de nuestros pares externos (sin exportar).

Una vez realizada esta configuración básica, el ISP puede iniciar la activación estableciendo una ruta estática para que el prefijo atacado (o host) sea conducido al blackhole, por ejemplo:

```
ip route 172.16.0.1 255.255.255.255
192.0.2.1 Null0 tag 199
```

La ruta estática mostrada con anterioridad es el *activador* que inicia el proceso de enrutamiento del blackhole. El router en el que se configura esta ruta le anunciará la misma a todos los enrutadores internos, incluidos los enrutadores edge, a través de un iBGP. Cualquier router con una ruta estática hacia la interfaz de desecho para la 172.16.0.1/32, inmediatamente hará desaparecer el trafico local.

Así mismo, el ISP podría establecer una activación automática a través del BGP, de manera que un cliente de BGP pueda activar la ruta del blackhole, independientemente de la intervención del ISP. Este es el rasgo más poderoso del enrutado activado del blackhole. La configuración del lado del ISP es ligeramente diferente en esas comunidades y se emplea un ebgp-multihop para recibir y etiquetar adecuadamente las rutas que aprenden los clientes. La configuración básica por parte del ISP aparece en el Listado 2.

El ISP ya tiene el <blackhole-ip> enrutado estáticamente para las interfaz de desecho a lo largo de la red, por tanto, tan pronto como el cliente anuncia el prefijo que debe ser conducido al blackhole, el ISP lo redistribuye internamente y el tráfico que viene a este prefijo se conduce al blackhole en el límite de la red ISP.

La configuración básica del cliente aparece en el Listado 3.

Una vez se configure el BGP, el cliente sólo necesita instalar una ruta estática para el prefijo # blanco del ataque. Con alguna configuración muy básica en el BGP, y con la ayuda de su ISP, ahora cuentas con un método muy rápido para responder a los ataques de denegación de servicio contra un host solamente, o contra un prefijo entero.

Nota: Asegúrate de verificar con el servicio técnico de tu ISP antes de llevar a cabo su solución de activación del blackhole, ya que la puesta en práctica de este concepto por parte de los diferentes ISP puede diferir ligeramente.

Los Backscatter y Tracebacks

En esta sección exploraremos algunos usos creativos de las redes de señuelo para detectar ataques y falsificaciones, así como para contribuir a localizar al autor de estos perjuicios.

El Backscatter

Después de todo lo que se ha hablado sobre las redes de señuelo y los ataques DDoS resulta propicio mencionar el concepto de backscatter o dispersión inicial. Durante un semestre entero de mi primer año de universidad, le escribí cartas (sí, de las convencionales) a varios amigos que se mudaban con frecuencia. Como soy muy despistado, a menudo escribía la dirección de devolución equivocada en el sobre. Se me olvidaba poner el número de mi habitación en la beca o lo escribía totalmente ininteligible (ya había descubierto la cerveza). De vez en cuando, alguno de mis amigos a los que había escrito se mudaba y la carta que le había enviado retornaba con una notificación de correo que decía *devolver al remitente*. Pero como mi dirección de devolución era incorrecta la carta no llegaba a mí sino a la oficina de residentes en el piso de abajo. Desde allí me llamaban (identificando mi nombre) para hacerme saber que nuevamente había escrito mal mi dirección y que tenían una carta allí en espera de que la recogiera para reenviarla. Ese *devolver al remitente* es una forma de backscatter o dispersión inicial. Claro, el backscatter indicaba a la oficina de residentes que yo había estado enviando correos (y a quien).

En Internet, cuando una parte A piensa realizar un ataque de denegación-de-servicio contra la parte B, pero la parte A quiere ocultar su identidad, normalmente escribe la dirección de la fuente equivocada en sus paquetes de ataque (los encabezamientos IP se falsifican para que parezca que salieron de las partes A-Z, por ejemplo, sólo de la A-Z en la IPv4 es de 2^{32} permutaciones). Durante tales ataques, los routers y otros dispositivos de red a lo largo del trayecto envían inevitablemente una serie de mensajes que van desde el restablecimiento de la conexión para satisfacer la solicitud hasta las notificaciones inalcanzables. Puesto que estos mensajes son *devueltos al remitente*, y ya que el remitente es falsificado, todas las partes de la A a la Z los reciben, y por tanto conocen del ataque a la parte B, de la misma manera en que la oficina de residentes supo de los correos que yo enviaba. Esto se muestra en la Figura 5.

En la actualidad cuando se trata de filtrar paquetes, la mayoría de estos mensajes del backscatter son desechados silenciosamente por los firewalls porque son vistos como respuestas a mensajes no enviados. No obstante, con una red de darknet externa puesta en práctica de la manera en que explicamos con anterioridad, podemos buscar estos paquetes de backscatter y determinar cuando nuestro espacio de dirección ha sido implicado en un ataque a otra parte. Los siguientes tipos de paquetes que aparezcan en la darknet pueden clasificarse como backscatters e indicar que su (darknet) espacio de dirección está implicándose en un ataque (ver Tabla 1).

El Traceback

Ahora que ya tenemos conocimientos sobre el backscatter explicaremos cómo utilizarlo. Dentro de una red con pasarelas múltiples de tránsito de Internet, resultaría útil localizar los puntos de ingreso de los *paquetes defectuosos* durante un ataque debilitado. Esta técnica, conocida como traceback (rastreo), es válida puesto que una vez que identifiquemos el punto específico de

ingreso en nuestra red (o en nuestro ISP), podemos estar en condiciones de disminuir el tráfico allí y reducir la carga en nuestros vínculos, e incluso podemos potencialmente permitir que *el tráfico válido* fluya (a través de pasarelas alternas); a diferencia de la táctica de protección contra el DDoS de blackhole que es más sencilla y que discutimos con anterioridad. El rastreo o Traceback nos permite utilizar el backscatter que recogemos en nuestro(s) darknet(s) como un medio para encontrar el punto donde el ataque está penetrando en la red. Desgraciadamente, esto sólo es viable para ISPs o redes de datos de largo alcance con muchas pasarelas de Internet. Algunas dependencias que van más allá de esta descripción incluyen la utilización del mecanismo de defensa del blackhole en cada pasarela de Internet. Dado que los principales ISPs hacen esto junto a un grupo de redes de compañías globales, resulta adecuado al menos explicar el proceso.

Si asumimos que tu red está configurada en correspondencia a lo anteriormente expuesto, puedes realizar un rastreo (traceback) en medio de un ataque de denegación de servicio en tres pasos fáciles:

- Identifica el blanco y verifica que el tráfico del ataque está siendo falsificado (si no es así, está táctica de rastreo será inútil).
- Establece un blackhole para la ruta de los hosts específicos (probablemente los /32s) que son atacados en cada una de sus pasarelas. Se cauteloso y usa las mejores prácticas con respecto a la utilidad de remitir hacia la interfaz de desecho en lugar de usar un filtro de paquetes para reducir los paquetes de ataque. Esta operación de blackhole hará que este router de pasarela comience a generar mensajes de ICPM inalcanzable, los cuales son (intentan ser) devueltos a las fuentes falsificadas de los paquetes de ataque.
- Dentro de tus darknets, utiliza las herramientas de darknet que has colocado para buscar el tráfico de backscatter (probablemente en forma de ICPM inalcanzables) con la dirección IP de tus routers de pasarela en su interior. Cualquier dirección IP de tu pasarela que veas como la fuente de estos paquetes de backscatter confirma que tales pasarelas son realmente el punto de ingreso del tráfico de ataque. Voilá, ha encontrado el lugar donde el ataque está penetrando la red. Aun cuando no tenga configuradas sus herramientas sofisticadas de darknet, una simple lista de acceso aplicada a la interfaz del enrutador puede funcionar, como se describe a continuación:

```
access-list 105 permit icmp any
any unreachables log; access-
list 105 permit ip any any
```

Por tanto, si accedes al modo de monitoreo terminal en esta lista de acceso (o si simplemente reduces el registro), conseguirás un informe pobre del backscatter, el que podrás estudiar para encontrar las direcciones IP de sus pasarelas. La táctica de traceback o rastreo y la defensa de blackhole contra los ataques de DDoS son útiles en situaciones donde las inundaciones de tráfico maligno han falsificado los encabezados. Ahora bien, con la proliferación de máquinas zombis y botnets, muchos atacantes han dejado de falsificar los paquetes DDoS completamente – no existe razón para falsificar encabezados si su ejercito de sistemas atacantes está por todos lados. De igual manera, los ataques DDoS falsificados han disminuido considerablemente como resultado de una mayor utilización del filtrado de ingreso y de uRPF. ●

En la Red

- Extreme Exploits: Advanced Defenses against Hardcore Hacks, publicado por McGraw-Hill/Osborne. Derecho de autor 2005 *http://www.amazon.com/gp/product/0072259558/*
- Internet RFCs 3330 (Special-use IPv4 Addresses) and 3882 (La configuración del BGP para bloquear ataques de denegación de servicio)
- The Team Cymru Darknet Project *http://www.cymru.com/Darknet/*
- The home of tcpdump and libpcap *http://www.tcpdump.org/*
- The home of ARGUS *http://www.qosient.com/argus/flow.htm*
- The home of Honeyd *http://www.honeyd.org*
- The home HoneyNet Project *http://www.honeynet.org*
- The home of the p0f tool *http://lcamtuf.coredump.cx/p0f.shtml*
- Artículo de Chris Morrow y Brian Gemberling sobre el proceso de blackhole del ISP y el análisis del backscatter: *http://www.secsup.org/Tracking/*
- Presentación de Dan Hawrylkiw sobre las honeynets. *http://phoenixinfragard.net/meetings/past/200407hawrylkiw.pdf*
- Preguntas más frecuentes acerca del filtro del paquete OpenBSD *http://www.openbsd.org/faq/pf/*

Sobre el autor

Victor Oppleman es un autor consumado, orador y maestro en el campo de la seguridad de las redes y es también asesor de algunas de las compañías más admiradas del mundo. El software de fuente abierta de Oppleman se ha distribuido a centenares de miles de computadoras en todo el mundo y posee las patentes estadounidenses de propiedad intelectual de enrutado adaptativo distribuido y de las aplicaciones inalámbricas del consumidor. La mayoría del contenido de este artículo ha sido extraído del libro del Sr. Oppleman, Extreme Exploits: Advanced Defenses Against Hardcore Hacks, publicado por McGraw-Hill/Osborne (Derechos de autor 2005) que seguramente está disponible en tu librería favorita.

Foco

Protección de IPv6

Rita Pužmanová 

Grado de dificultad



IPv6 es el protocolo de red IP de nueva generación que no podemos simplemente dejar de lado. Cada vez hay más presión para aceptarlo. Nada pasa tan casualmente – IPv6 ofrece una serie de ventajas indiscutibles frente a su antecedente. Este protocolo tiene no solamente un amplio espacio de direcciones sino que también trae incorporado soporte para las soluciones inalámbricas, aplicaciones dispersadas y seguridad.

Aunque la mayoría de los sistemas operativos actuales y dispositivos de red ya están dotados de soporte IPv6 (*Internet Protocol version 6*), sin embargo, el uso real de este protocolo en las redes todavía no es universal. En ello influyen muchos factores, sobre todo el coste relacionado con el cambio de IPv4 a IPv6 e insuficiente conciencia de las ventajas que lleva el segundo. El presente artículo está destinado a una de las ventajas de IPv6 frente su antecedente Ipv4, que es seguridad integrada.

Protecciones IPv4

Las protecciones de IPv4, igual como las de IPv6, se encuentran en permanente fase de desarrollo, lo que siempre lleva cierto riesgo. Algunos problemas como las protecciones de las aplicaciones móviles o multicast no las analizaremos, ya que requerirían demasiado detalle. Nos convenceremos de que la protección es solamente uno de las ventajas que ofrece a los usuarios el cambio a este protocolo de nueva generación. IPv6 ofrece direcciones únicas a los diferentes dispositivos terminales o sensores. Permite también movilidad y comunicación real *peer-to-peer.*

La seguridad en IPv6 se parece mucho a la seguridad en IPv4. Lo que esperamos sobre todo de la seguridad a nivel de red es saber quien nos envía mensajes, quién lee los mensajes reenviados por nosotros y si los mensajes no son modificados durante la comunicación. Además, es necesario que la red sea funcional y accesible para los usuarios autorizados y que tengamos control sobre nuestro propio dispositivo conectado a Internet.

En este artículo aprenderás...

- de cómo evaluar los rasgos de IPv6 y las ventajas de su aplicación,
- de si es necesario decidir de su implación,
- de cuales son las amenazas básicas de IPv6 y qué medios de defensa tenemos.

Lo que deberías saber...

- deberías conocer las bases de TCP/IP, sobre todo direccionamiento IPv4 (eventualmente IPv6),
- deberías conocer las protecciones que existen en las redes IP, sobre todo IPSec.

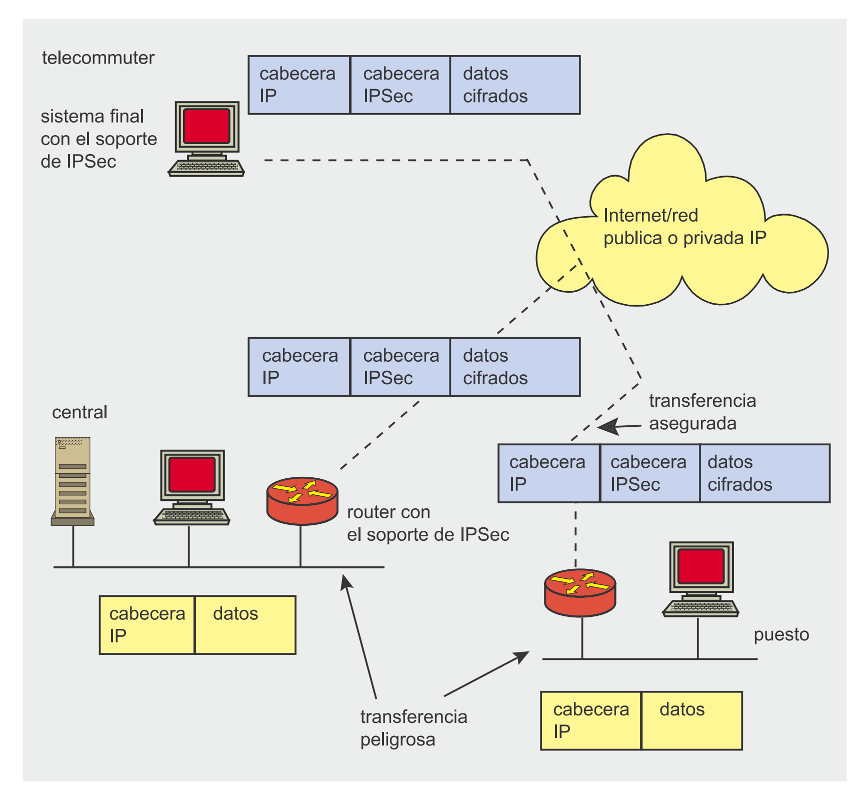


Figura 1. *Comunicación por la red pública con IPSec*

IPSec: arquitectura de la protección de Red

Primero observaremos la arquitectura de seguridad que emplean ambos protocolos. La arquitectura de seguridad IP (*IPSec,* RFC 4301) tiene como objetivo una fuerte protección para IPv4 y para IPv6. La arquitectura IPSec fue diseñada en realidad por primera vez para IPv6 (RFC 1883). IPSec soporta la autenticidad, integridad y ocultación a nivel de datagrama. Toda la arquitectura está compuesta por unos protocolos que sirven para enviar los datos autenticados o cifrados en las redes IP (ver la Figura 1).

La descrpción completa de IPSec se encuentra en hakin9 n° 16 (*IPSec: Descripción técnica* por Benoni Martin).

Protocolos AH y ESP

La autenticación e integridad de mensajes IP están aseguradas por medio de un suplemento seleccionable al datagrama IP, en forma de la cabecera de autenticación (*Authentication Header,* RFC 4302) introducida en lugar de una cabecera original IP, que emplea el cifrado por medio de la clave pública. El cifrado se aplica a todas las partes del datagrama que durante el proceso no cambian. Aquí se emplea el algoritmo de cifrado *MD5*. El cifrado se realiza al principio antes de desfragmentar el datagrama y el descifrado se realiza después de componer una vez más el datagrama en la estación de destino.

La autenticación de AH se realiza por medio del así denominado código de autenticación de mensajes (MAC, *Message Authentication Code*). Cuando empleamos un resumen unilateral el resultado es HMAC (*Hash-Based MAC*); surgió como resultado de la combinación de una función de resumen segura con función de cifrado. Podemos resumir cualquier texto, recibiendo la serie final de resumen de determinada longitud. A diferencia de las firmas digitales, al cifrar la serie final de resumen empleamos la clave privada de la misma longitud. HMAC emplea la función MD5 (RFC 2085 y 2403) o de otra más fuerte y más exigente SHA-1 (RFC 4305). AH es el método respectivo donde es suficiente la autenticación de cada datagrama por separado. AH está dotado de la identificación del protocolo protector, número de orden y valor de autenticación (ver Figura 2).

El segundo protocolo IPSec, *Encapsulating Security Payload* (ESP, RFC 4303), incluye mensajes secretos por medio del cifrado del contenido de los datos y de la cabecera, compartiendo además unos servicios parecidos como AH (la comparación de AH y ESP se encuentra en la Tabla 1). ESP es adecuado en los casos más exigentes, cuando es necesario autenticar y cifrar el contenido de los datagramas con el fin de proteger los datos contra escucha o uso excesivo. ESP permite cifrar junto con autenticación o bien solamente la autenticación (*null encryption,* RFC 2410). ESP define el posible contenido del datagrama IP. Está compuesta por la cabecera con información relacionada con la protección del protocolo (SPI) con el número de orden, eventualmente con información requerida por un algoritmo de cifrado (por ejemplo DES), ver la Figura 3. Los datos se cifran respectivamente y después de terminar el datagrama

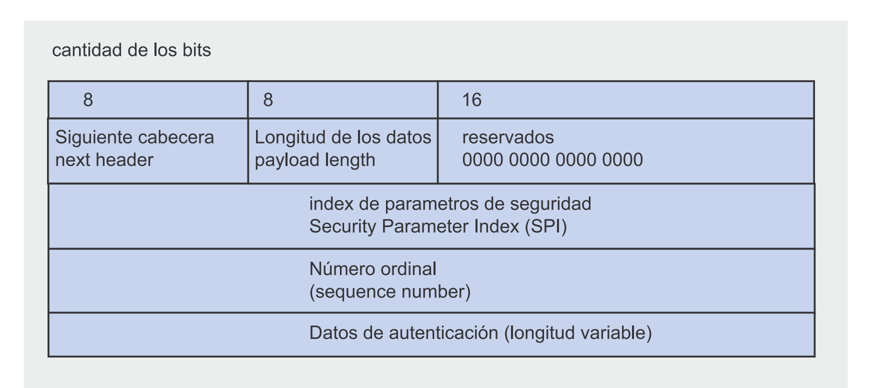
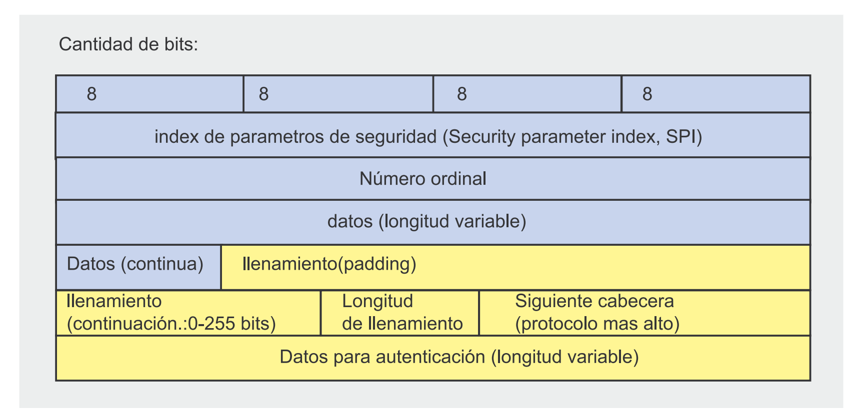


Figura 2. *Formato de la cabecera autenticada (AH)*


Cantidad de bits:

8	8	8	8
index de parametros de seguridad (Security parameter index, SPI)			
Número ordinal			
datos (longitud variable)			
Datos (continua)	llenamiento(padding)		
llenamiento (continuación.:0-255 bits)	Longitud de llenamiento	Siguiente cabecera (protocolo mas alto)	
Datos para autenticación (longitud variable)			

Figura 3. *Formato de cabecera y final (cifrado) ESP*

se da la suma de control para comprobar la corrección del datagrama.

AH y ESP emplean una función de resumen para comprobar que, durante la comunicación en red, ha tenido lugar el intercambio de paquetes original. Teniendo en cuenta el hecho de que hay cambios de cabecera, ambos protocolos añaden valor al control de integridad (ICV*, Integrity Check Value*) a una parte de las cabeceras que pertenecen a las capas superiores.

Modos de túnel y de transporte

La asociación de seguridad funciona en dos modos: de transporte y de túnel (Figura 4). En el modo de transporte la cabecera de protección se introduce entre la cabecera original del datagrama IP y los datos, mientras que en el modo de túnel se produce una nueva cabecera para el datagrama detrás del cual se introduce una cabecera de protección más.

ESP en el modo de transporte cifra de forma selectiva los datos transportados pero no autentica la cabecera del datagrama. AH en el modo de transporte autentica los datos transportados y los campos seleccionados de la cabecera del datagrama. En el modo de transporte el valor del campo *siguiente cabecera* del datagrama IP son *51* para AH y *50* para ESP lo que equivale a la correcta identificación de protocolo. Ambas cabeceras incluyen también el campo que indica la siguiente cabecera con la especificación o bien se trata de los datos TCP ó UDP. El modo de transporte es adecuado para la protección de comunicación final entre las estaciones, realizada por medio de una red externa (ver la Figura 5).

En el caso del modo de túnel todo el datagrama original (interno) se empaquetará sin cambiarse en otro datagrama (es decir: se cifrará libremente antes de empaquetar) con cabecera no cifrada (datagrama externo) que servirá para determinar una ruta determinada en la red. La cabecera del datagrama original da la información sobre el fin del datagrama, mientras que la cabecera IP externa (nueva) determina el final del túnel. Esta forma de comunicación funcionará solamente entre los routers que cooperen entre sí y eventualmente entre las estaciones finales configuradas para el propósito. Los routers encaminan solamente usando la cabecera del datagrama externo. El modo de túnel se aplica por defecto en la construcción de una VPN (*Virtual Private Network*; ver la Figura 6).

El empleo de ESP en modo de transporte y en el modo de túnel – teniendo en cuenta las partes del datagrama que se autentican y las que son cifradas – se describe en la Figura 7.

Podemos emplear los mecanismos ESP y AH entre cualquier nodo en red (entre los usuarios finales o entre los routers), en el modo de comunicación individual o emisión en grupo. Los mecanismos de protección los podemos unir entre sí, lo que significa que el datagrama puede incluir ambas cabeceras: tanto AH como ESP. AH asegura la integridad de los datos en el modo sin conexión autenticación del origen IP de paquetes, sin embargo no ofrece la ocultación de los datos por medio del cifrado. ESP permite cifrar, sin embargo, no protege las nuevas cabeceras IP del paquete original hermetizado. Para conseguir autenticación fuerte junto con la ocultación de información transportada, será necesario emplear AH en la combinación con ESP en ambos modos: tanto en el de transporte como en el de túnel.

Traducción de direcciones: NAT

La traducción de direcciones (NAT, *Network Address Translation*; RFC 3022) se emplea frecuentemente en la práctica, que en IPv4 tiene dos causas: delimitar la cantidad de direcciones IP únicas para las redes privadas y mejorar la protección de las comunicaciones entre las redes privadas e Internet. NAT es una solución temporal en la situación de un espacio insuficiente e ineficaz de direcciones IPv4 – las direcciones están limitadas por la longitud de 32 bits. Tal situación provisional puede transcurrir hasta pasemos a una nueva – sexta versión del protocolo IP, que permite el único direccionamiento de un mayor número de los nodos de red. Estas direcciones tienen la longitud de 128 bits.

La función de NAT es conocida generalmente, así que la recordare-

Tabla 1. *Comparación de protocolos AH y ESP*

	AH	ESP
autenticación de la fuente de mensaje	sí	a seleccionar
integridad de migración de los datos	sí (incluyendo la cabecera)	sí (además de la cabecera)
defensa contra el ataque replay	a seleccionar	sí
(ocultación de los datos transportados (cifrado)	no	sí

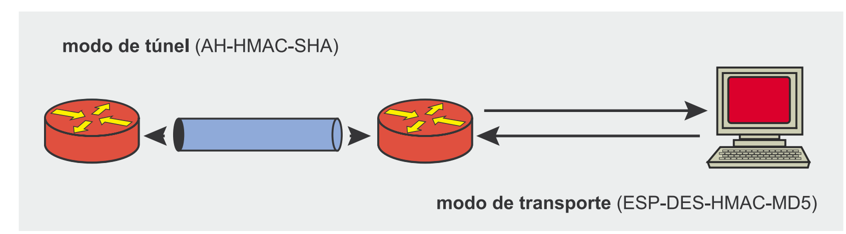


Figura 4. *Protección de asociaciones en dos modos*

mos sólo brevemente. Sobre todo debemos recordar que la red de conexiones a Internet por medio de NAT debe incluir al menos una dirección IP importante globalmente que está asignada al router o a la estación con la conexión a Internet. El dispositivo NAT (así denominado *NAT box*) traduce las direcciones de los datagramas de entrada y de salida de tal forma que sustituye la dirección de procedencia en los datagramas de salida a la dirección global e importante y la dirección de destino en los datagramas de entrada con una dirección privada de la estación dada de destino (según RFC 1918). Desde el punto de vista del usuario externo todos los datagramas vienen desde el dispositivo NAT y las respuestas salen en su dirección. Desde el punto de vista del usuario interno, el dispositivo NAT es visible como router que tiene conexión a Internet.

Disgustos a causa de NAT

En numerosas aplicaciones la regla del NAT no se puede aceptar, ya que estas aplicaciones no pueden cooperar correctamente en la situación de traducción de direcciones. NAT no puede cooperar con protocolos que emplean información relacionada con las direcciones dentro de datagramas independientes (las partes de aplicaciones o de transporte de los datos de control), pero no solamente en la cabecera IP del datagrama. En la cabecera se encuentran las direcciones copiadas por medio de NAT, pero dentro de los datagramas NAT ya no realiza su operación del cambio de dirección. La nueva versión de NAT es capaz de solucionar este problema. NAT en general no está afinada a numerosas intenciones de las funciones de IP, por lo tanto no podemos esperar que todas las aplicaciones funcionen en la presencia de NAT tan eficazmente como fueron pensadas. Podría parecer recomendable que todas las aplicaciones tolerasen NAT, sin embargo, tal tolerancia constituye frecuentemente una barrera importante teniendo en cuenta el rendimiento, la capacidad de escalar y la implementación de aplicación.

NAT ocasiona problemas incluso para IPSec, donde existe la necesidad de abrir comunicación entre dos direcciones finales. Esto no es comparable con la situación cuando para la comunicación se incluye una dirección substitutiva. IPSec al emplear la cabecera autenticada cuenta el valor de autenticación de todo el datagrama – incluyendo las direcciones IP de procedencia y de destino. Cualquier cambio de dirección IP, por ejemplo, por medio de la traducción de dirección conduce inevitablemente a contar otro valor, con lo cual falla la autenticación. Por lo tanto, es necesario emplear NAT antes de utilizar de IPSec. En el caso de ESP la autenticación no se realiza con una cabecera externa, con lo cual, en caso de NAT puede tener lugar después de IPSec.

En el caso de una combinación de traducción de direcciones y puertos (*NAPT, Network Address and Port Translation*) en realidad se usa solamente una dirección IP, sin embargo, hay más puertos TCP y UDP (objetivo: ahorrar direcciones). NAT supone los valores de puertos detrás de la cabecera IP en la unidad de datos. Con el empleo de IPSec la suposición anterior no puede cumplirse. Además ESP cifra la cabecera TCP o UDP aún en el modo de túnel lo cual para NAT será un problema; por lo tanto la traducción de las direcciones debería tener lugar antes de aplicar IPSec. Una serie de los nuevos productos IPSec soporta el empleo de NAT aplicando hermetización de UDP, sin embargo, no es una solución universal.

Protocolo de la nueva generación de IP versión 6 y su seguridad

La nueva versión del protocolo IP (*IPng, IP next generation*), con el número 6 (IPv6; RFC 2460) fue elaborada ya hace más de diez años. La necesidad de modernizar la IPv4 existente está relacionada con el insuficiente espacio de direcciones y su empleo no sistemático, muchas veces con el uso abusivo de de la asignación de bloques de direcciones demasiado grandes. IPv6 soluciona los susodichos problemas gracias al formato que permite el empleo de 1038 direcciones únicas, sin embargo, su empleo en las redes presentes está motivado no solamente por otras premisas que sólo el mayor espacio de direcciones.

Gracias al direccionamiento individual, IPv6 permite la comunicación completa sin intermediarios entre los dispositivos finales en forma de *peer-to-peer.* Sin duda alguna mejor que el anterior soporta los servicios y aplicaciones modernos y los que surgen nuevamente tales como VoIP, juegos online llevados por un gran número de participantes, vídeoconferencias, servicios móviles relacionados con la

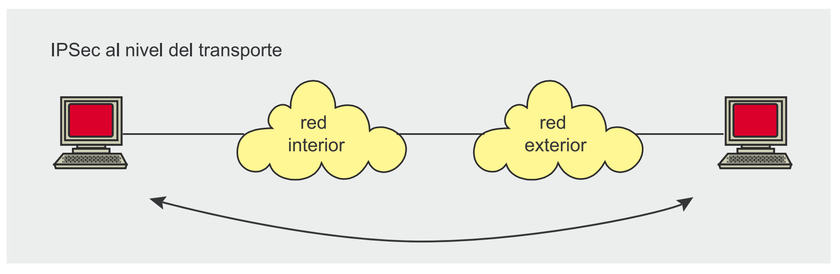


Figura 5. *Empleo del modo de transporte*

transferencia de datos así como también la conexión remota de sensores (por ejemplo RFID, *Radio Frequency IDentification*), casas informatizadas y edificios con inteligencia artificial o bien *grid computing*.

A pesar de ello no podemos hablar de una arquitectura de red totalmente nueva, ya que IPv6 heredó una serie de rasgos de IPv4. Es el mismo servicio de datagrama, hay idénticos protocolos de transporte – e incluso unas aplicaciones prácticamente iguales. IPv6 ofrece, sin embargo, también una serie de nuevos elementos, entre otros, más direcciones, autoconfiguración, soporte de movilidad y seguridad integrada.

Direccionamiento en el protocolo IP versión 6

Para cumplir con los requisitos relacionados con la extensión y la jerarquía de las direcciones para IP, IPv6 emplea 128 en vez de 32 bits (RFC 4291). El espacio de direcciones es, por lo tanto, enorme: 2128. Una dirección IPv6 está dividida típicamente en dos partes: el prefijo y la identificación de la frontera. Una frontera de red puede tener incluso direcciones IPv6. En IPv6 destacamos direcciones individuales (*unicast*; RFC 3587), en grupos (*multicast*; RFC 3306, 3307 y 3956) y asignadas a cualquier grupo (*anycast*) – con lo cual solamente los dos primeros tipos se emplean en IPv4. El nuevo tipo de dirección *anycast* es muy útil para una serie de aplicaciones modernas que emplean servidores situados geográficamente en Internet. Por otra parte, las direcciones reservadas pueden indicar al intruso un objetivo interesante – por lo tanto las direcciones *anycast* accesibles globalmente deberían definirse solamente para los routers. Para la dirección de destino *anycast* no existe ningún mecanismo de autorización, lo que facilita los ataques de tipo *spoofing* y *masquerade*.

Las direcciones tienen dos funciones básicas que en IPv4 coexisten y las que IPv6 trata de separar – se trata de la función de situación y de identificación. La información sobre la situación (localizador) es necesaria al router en la red ya que determinan la forma de encontrar el camino al objetivo. Ofrecen los mínimos tres niveles de agregación (*TLA, Top-Level Aggregator; NLA, Next-Level Aggregator* y *SLA, Site-Level Aggregator,* RFC 3587).

La Identificación (identificador) indica un dispositivo en concreto o su interfaz. La agregación o la asignación jerárquica de direcciones se emplea muy eficazmente para el direccionamiento eficaz de datos, cuando en el camino de red se toman decisiones gradualmente, a partir de los bits iniciales de dirección, se toman decisiones según los prefijos más largos.

En IPv6 los primeros 64 bits indican información sobre la localización, para que sea posible alcanzar cierto *site*. Los siguientes 64 bits muestran la identificación del dispositivo en el alcance especificado. El paso a un nuevo proveedor de servicios de Internet requiere cambio del localizador, pero no de identificación. Las direcciones Ipv6 jerárquicas más largas satisfacen sobre todo encontrar eficazmente el camino del paquete en red, ya que permiten una agregación más fácil de las direcciones según los niveles jerárquicos en red, del proveedor de servicios de Internet (conexión), de la empresa que emplea Internet etc. (RFC 3587).

La notación de las direcciones IPv6 se diferencia de las de IPv4. Las direcciones se guardan aquí en forma hexadecimal y los respectivos pares de octetos (de cuatro cifras) están separados por dos puntos. Una dirección puede ser, por ejemplo, la siguiente: FBCD:1234:5678:9001:2222:AB12:2345:6789. Los grupos nulos se pueden eliminar de la inscripción y se pueden sustituir con un par de dos puntos, sin embargo, solamente una vez por dirección, en caso contrario podría tener lugar una ambigüedad. Por lo tanto una dirección como FEDC:0000:0000:0000:0000:0000:0000:0110 la podemos expresar como FEDC::110. RIPE NCC (*Réseaux IP Européens Network Coordination Centre*) tiene asignado de IANA (*Internet Assigned Numbers Authority*) un bloque de direcciones (2001:600::/23). Aunque sumemos todos los tipos de direccionamiento en IPv6, obtendremos un listado bastante largo. IPv6 permite la asignación a una interfaz de red muchas direcciones individuales que pueden ser globalmente unívocas (global), solamente localmente (site-local) o bien para la red dada (link-local). Las direcciones dentro de IPv6 se pueden categorizar según el estado de configuración (RFC 2462) en preferidas (preferred) y depreciadas (deprecated); eventualmente conforme con RFC 3041 en públicas (public addresses) y provisionales (temporary addresses).

La conexión a un mayor número de proveedores de Internet, es decir, multihoming, conduce a la asignación de un mayor número de direcciones a un nodo – también para las interfaces virtuales o interfaces de túnel. Por lo tanto, al iniciar una comunicación, las implementaciones de IPv6 frecuentemente se encuen-

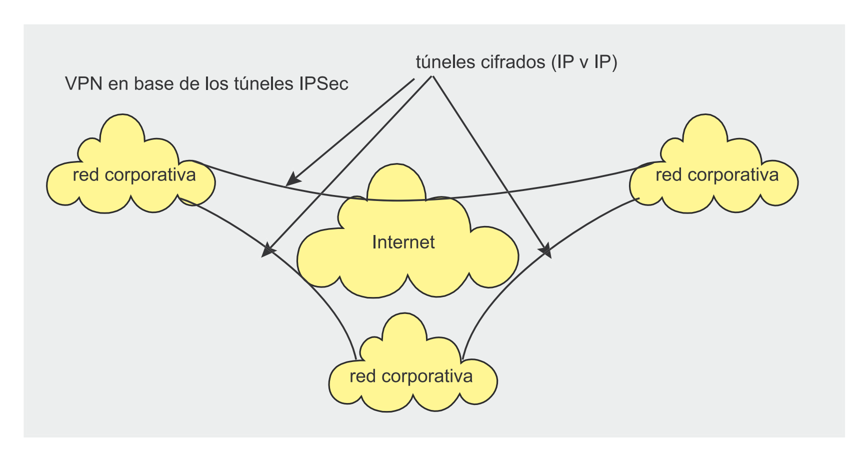


Figura 6. *Empleo del modo túnel*

tran con la necesidad de seleccionar en los nodos y routers entre unas direcciones de procedencia y de destino. Tales situaciones se resuelven por medio de un mecanismo indirecto de direcciones (RFC 3484), común para todas las implementaciones el que, sin embargo, no tiene preferencia en caso de seleccionar direcciones concretas para las respectivas aplicaciones. En el caso de las implementaciones duales (IPv4 e IPv6) es necesario decidir el tipo de dirección que emplearemos.

Configuración automática

IPv4 no ofrece configuración automatizada directamente de la estación durante su conexión a Internet. Una alternativa para la configuración manual de cada estación es el protocolo DHCP. A base de la pregunta por parte de una estación, el servidor DHCP le asignará una dirección IP y suministrará el resto de información necesaria para el trabajo en la red dada. El protocolo adaptado DHCPv6 se puede emplear para Ipv6. Sin embargo, una aportación positiva de IPv6 es la configuración automática (RFC 2462 y RFC 3041) que no requiere ningún servidor.

La configuración automática no está incluida en el proyecto IPv6 a causa de *la falta de confianza* de los usuarios finales sino para facilitar los cambios de ISP, soporte de movilidad, aseguración direcciones unívocas y soporte de IP en los dispositivos en los cuales no hay ningún administrador de red (se trata, por ejemplo, de un dispositivo de recepción de casa). La configuración automática sin estado permite también a los nodos la comunicación en las redes sin routers (redes *ad hoc*).

La configuración automática emplea el protocolo ICMPv6 (*Internet Control Message Protocol*) y se basa en el *reconocimiento de vecinos* (NDP, *Neighbor Discovery Protocol,* RFC 2461 y RFC 3122). La estación que está conectada a la red IPv6, primero creará su dirección local (*link-local*) a partir de un prefijo definido de forma preliminar *FE80*, y luego añadirá su identificador (EUI*, End User Identifier*). La estación verificará en la red que nadie más tenga esta dirección. Todos los nodos en el segmento dado contestarán a la estación dada y después de intercambiar esa información podrán comunicarse mutuamente sin el empleo de servidores o routers.

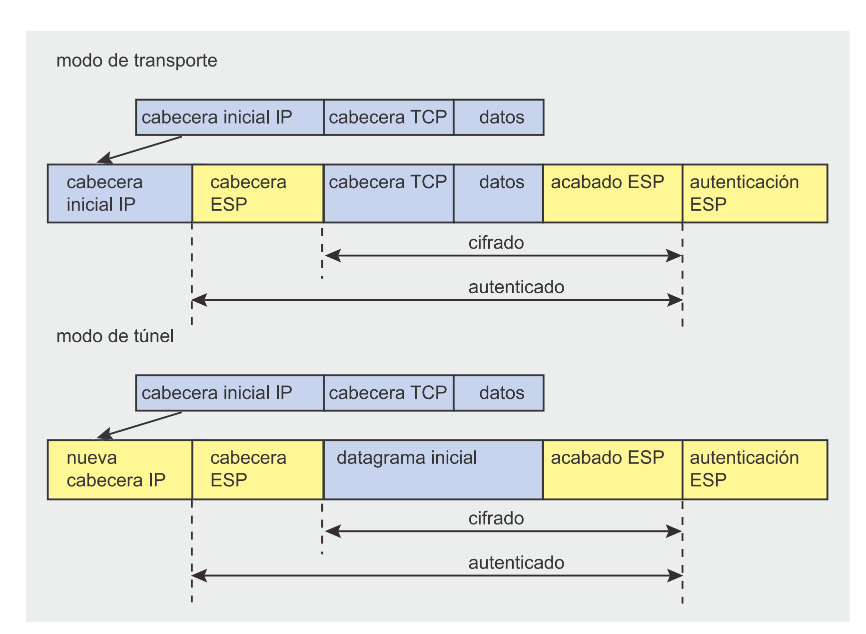


Figura 7. *ESP en el modo de transporte y de túnel*

Las estaciones monitorizan los mensajes de routers que de manera regular envían información (*router advertisement*) informando a las estaciones sobre el *prefix adress* de la red dada y la información relacionada con el router de acceso (*default gateway*) y su estabilidad. Al mismo tiempo, gracias a esta información la estación se sabrá si emplear la configuración con determinación del estado o bien sin determinarlo. Una estación conectada de nuevo puede por si misma pedir información de los routers, con lo que no tiene que esperar su anuncio periódico. Dentro de la configuración sin estado, la estación generará su propia dirección unívoca IPv6 al añadirla al prefijo anunciado EUI de la dirección local. Si la estación tiene como objetivo realizar la configuración con estado, se empleará DHCPv6.

Teniendo en cuenta la seguridad es necesario verificar que la información anunciada en la red viene del router autorizado y cuáles son los requisitos en cuanto a la protección relacionada con el ya mencionado *reconocimiento de vecinos*.

SEND (*SEcure Neighbor Discovery*; RFC 3971) define las nuevas posibilidades de ICMPv6 en cuanto al protocolo NDP a base de firma con claves públicas. El valor de la función de resumen de la clave pública se emplea para generar la dirección, los routers se certifican por medio de X.509, los datos se firman y los tags de tiempo confirman el momento de emisión del mensaje.

Datagrama IP en la versión 6

IPv6 mueve a extensiones opcionales de las cabeceras información seleccionada, cuya presencia en el datagrama no es necesaria. La *cabecera obligatoria* (ver la Figura 9) tiene longitud fija (40 octetos) e incluye solamente ocho campos (para comparar la Figura 8 se da el formato del datagrama IPv4). Detrás de la cabecera obligatoria pueden aparecer *cabeceras opcionales* de longitud variable. Los datos que incluye se emplearán después en los nodos finales y pocas veces en los routers. Gracias a ello los routers se ocupan solamente de la cabecera de longitud fija menos complicada que IPv4, lo que permite acelerar su trabajo.

A diferencia de IPv4, la cabecera obligatoria no incluye aquí demasiada información sobre la longitud de la cabecera y, al mismo tiempo tampoco incluye la suma de con-

trol, incluyendo otras capas. IPv6, a diferencia de IPv4, no permite a los routers intermediar en la transmisión para fragmentar el datagrama según la longitud admitida en las unidades de datos para la red dada (MTU*, Maximum Transmission Unit*). Esto significa que la MTU mínima del camino dado debe confirmarse por la estación fuente (RFC 1981) antes de enviar el datagrama. Prohibir la fragmentación de datagramas por los routers delimita hasta cierto grado la posibilidad de un uso excesivo de la fragmentación con el objetivo de violar la seguridad de transmisión.

Detrás de la cabecera obligatoria IPv6 pueden aparecerer algunas de las cabeceras opcionales de extensión. Cada una de las cabeceras identifica lo que tiene lugar detrás de ella – incluye el campo que indica *el tipo* de cabecera que aparece después (ejemplo en la Figura 10). Aunque no aparezca ninguna de las cabeceras, se especifica el protocolo de transporte por medio del *número de protocolo* que muchas veces es igual que en IPv4 (*6* para TCP, *17* para UDP, o bien *46* para RSVP, sin embargo *58* para ICMPv6). La identificación *59* significa *falta de la siguiente cabecera* – si allí aparece algún dato, debe ignorarse.

Seguridad de IPv6

IPv6 emplea obligatoriamente el marco de protección IPSec, lo que significa que nativamente soporta el cifrado, autenticación, VPN. Junto con el nuevo protocolo aparecen también unos ataques nuevos, por lo tanto no podemos esperar que IPv6 sea supersegura. Sobre todo, debemos darnos cuenta de que IPSec cuida de las protecciones de la capa de red y no de las aplicaciones por separado, con lo cual, de ninguna forma evita sus ataques. De manera parecida, en el caso de IPv6, no protegerá contra los ataques en forma de muchos datagramas.

Los mecanismos de autenticación y de ocultación se añaden por medio de las cabeceras seleccionables de extensión del datagrama IPv6. El soporte de estas cabeceras es obligatorio pero IPv6 no las define por medio de aplicación, por lo tanto, de la aplicación y del usuario precisamente depende si se encuentran las protecciones al nivel adecuado. No podemos declarar que IPv6 es indirectamente más seguro que su antecedente – podemos decir que IPv6 es un paso hacia la extensión de la autenticación mutua y protección en el nivel de transmisión. AH protege la integridad de datos (MAC) y la autenticación (comprobación de la identidad de fuente), sin embargo no protege la ocultación. El cálculo MAC se realiza en el origen antes de fragmentar el datagrama, de manera parecida al control de integridad hasta después de volver construir el datagrama con la estación de destino. MAC se refiere a todas las partes del datagrama que no cambian en el camino (como en el caso de los tipos de cabeceras). Para MAC empleamos MD5 y SHA-1.

ESP protege con el cifrado sólo los datos. Podemos cifrar o bien la parte de transporte de datos, es decir el segmento TCP/UDP o bien el mensaje ICMP (*transport-mode*) o bien un datagrama completo IPv6 (*tunnel-mode*; ver Figura 11). En el primer caso la estación origen se ocupa del cifrado y los routers *en el camino* se interesan solamente por las cabeceras no cifradas, obligatorias del datagrama IPv6 y de las cabeceras de extensión no cifradas. En caso del modo de túnel tiene lugar el cifrado de todo el datagrama (interno) y el empaquetado en otro datagrama con la cabecera no cifrada (datagrama externo). Este método funciona solamente entre los routers que cooperen y no entre los nodos finales. Los routers en el camino se basan solamente en la cabecera del datagrama externo.

Podemos juntar libremente los mecanismos de protección, lo que significa que un datagrama IPv6 puede incluir ambas cabeceras *AH* y *ESP* (ver Figura 12.). Si bien ciframos antes de autenticar, todo el datagrama se autentica solamente con las partes cifradas y no cifradas; primero, cuando realizamos la autenticación, la cabecera se inserta en el datagrama interno y este se cifra completo. Esta solución no es demasiado afortunada ya que perdemos la ventaja de tener la autenticación.

La protección IPv6 no asegura protección contra la denegación de la procedencia del mensaje (*non-repudiation*), el ataque de copiar el mensaje (*replay*) y sobre todo contra

Formato de la cabecera obligatoria

- versión (version) – número de la versión del protocolo (6),
- prioridad (priority) – permite la fuente de identificación de la prioridad de cada datagrama en comparación con el resto de los datagramas de la misma fuente, la prioridad desde el punto de vista de transmisión y suministro,
- *etiqueta del flujo de datos* (*flow label*; RFC 3697) – significa datagramas que requieren atención especial durante el encaminamiento; las estaciones y routers que no soportan este campo, no pueden cambiarlo. El flujo de datos se define como una serie de datagramas enviados a un remitente o un grupo de remitentes, de los cuales la estación de procedencia requiere un procedimiento especial,
- *longitud de datos del datagrama* (*payload length*) – longitud de la parte restante de del paquete IPv6, es decir la longitud de todas las cabeceras que completan la longitud del campo de datos,
- *siguiente cabecera* (*next header*) – identifica el tipo de la cabecera directamente detrás de la cabecera obligatoria IPv6 del datagrama,
- *máximo número de routers* (*hop limit*) – número admitido de routers (analogía del campo del período de duración en IPv4 – TTL*, Time To Live*); cada router reducirá una unidad. Si el valor se llega hasta *0*, no podemos pasar el datagrama y debe generarse el mensaje ICMP,
- *dirección origen* (*source address*) – dirección del origen de 128 bits,
- *dirección de destino* (*destination address*) – dirección del destinatario de 128 bits (en algunos casos no debe tratarse de la estación de destino si se empleó la cabecera extendida para direccionar).

el ataque de tipo *denial-of-service* (DoS).

El mecanismo *NAT* extendido en las redes IPv4 dentro de IPv6 no es necesario, aunque su analogía existe. Se trata de soportar la traducción de las direcciones IPv4 e IPv6 (RFC 2766). Este mecanismo incluye la forma de traducción del formato de la cabecera entre ambos protocolos, por lo tanto, en el nombre aparece la traducción de protocolos (*PT, Protocol Translation*).

¿Paso a IPv6?

Apareció el miedo de pasar a una red que sigue aumentando basada en Ipv4, que requiere cambio administrativo de numeración (la redirección completa de IPv4 en las direcciones IPv6) y costosos cambios de dispositivos y aplicaciones. Por lo tanto, junto con el desarrollo de IPv6 se iniciaron las pruebas para emplear de la mejor manera posible el espacio de direcciones IPv4. Así aparecieron soluciones extendidas hoy día en forma de:

- *traducción de direcciones de red* (NAT) – para delimitar el empleo de direcciones que se reconocen globalmente, en las redes finales,
- *direccionamiento de subredes con máscara de longitud variable* (VLSM) – una dirección de red IPv4 se puede dividir en unas subredes (*subnetting*) y luego, dentro de ellas direccionar las estaciones finales; antes de VLSM existía la posibilidad de tal direccionamiento, sin embargo, estaba limitada con la selección fija de la cantidad de bits que estaban destinados para direccionar subredes (largo de la máscara). Sin embargo, VLSM eliminó esta limitación y permitió el redireccionamiento eficaz de las subredes con numerosas estaciones finales (Ethernet y otras redes que emplean los medios de transmisión) y conexiones de dos puntos que necesitan dos direcciones para los nodos finales,
- *routing teniendo en cuenta la clase de direcciones IPv4* (CIDR) – el routing de protocolos modernos, sobre todo en BGP (entre los dominios o sistemas autónomos), no se dirige por medio de las fronteras de las clases de direcciones pero permite la eficaz agregación de las direcciones a los prefijos (*supernetting*). Con ello se reduce la cantidad de las inscripciones en la tabla de routers fronterizos y se simplifica y acelera su trabajo. Una pausa inmediata en el sentido hacia las direcciones de destino dentro del bloque agregado de direcciones no implica la necesidad de calcular toda la tabla de dirección.

Tabla 2. *Cabeceras de extensión y sus identificaciones*

identificación	cabeceras de extensión
0	hop-by-hop
43	routing
44	fragment
50	encapsulating security payload header, ESP
51	authentication header, AH
59	no next header
60	destination options
62	mobility

Orden de apariencia de las cabeceras opcionales

El listado de cabeceras y sus identificaciones se presentaron en la Tabla 2:

- *cabecera de opción paso a paso* (*hop-by-hop options header)* – define las posibilidades especiales que requieren la cooperación de cada router, por ejemplo, la advertencia sobre contenido interesante del datagrama (RFC 2711, selección con la identificación *5*),
- *cabecera de opción de la estación de destino* (*destination options header*) – incluye información opcional para la estación de destino o todos los objetivos según la cabecera de dirección en cuanto a la estación de destino es necesario rellenar para nivelar el datagrama.
- *cabecera de encaminamiento* (*routing header*) – comparte información de encaminamiento, tal como las direcciones de routers que deben intermediar en la transmisión, la estación de destino debe luego emplear esta cabecera para determinar el camino en sentido contrario,
- *cabecera de fragmentación* (*fragment header*) – incluye información para fragmentar y para volver a componer los datagramas. Sin embargo, la fragmentación puede hacerse solamente por medio de la estación origen, por lo tanto la estación origen debe ser capaz de fijar el valor máximo admitido de la unidad de datos en el camino hacia la estación de destino para que el datagrama no se destruya durante la transmisión. Volvemos a reconstruir el datagrama en la estación de destino,
- *cabecera de autenticación* (*AH*) – asegura la integridad y autenticidad del datagrama,
- *cabecera de la protección de la hermeticidad de los datos* (*ESP*) – asegura la protección de los datos transportados en el datagrama junto con la garantía de la protección de autenticidad e integridad de datos.

Las soluciones susodichas no eliminaron, sin embargo, la necesidad de pasar a IPv6: NAT ocasiona problemas provocados por la instalación de los mecanismos de protección en la red (IPSec) sin permitir también la comunicación directa *peer-to-peer.* CIDR no es suficiente para delimitar el crecimiento de las tablas direccionales en los routers reales.

Comparación detallada de IPv6 con IPv4

Las diferencias de los servicios IPv4 y de IPv6 se encuentran en la Tabla 3. Las diferencias que resultan de las características se reflejan en los formatos del datagrama IPv4 e IPv6 y se dan en la Tabla 4 (ver también las Figuras 8 y 9).

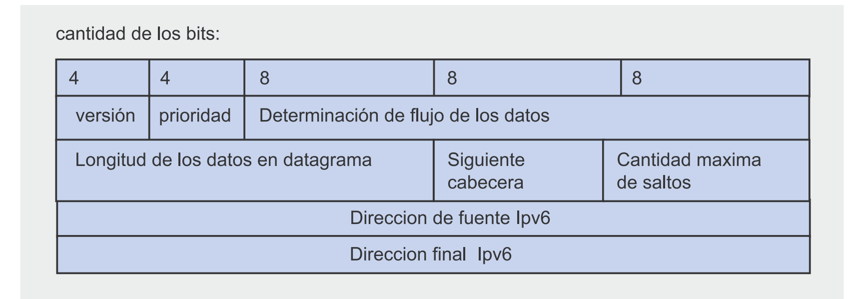
Paso de IPv4 a IPv6

Las implementaciones completas de IPv6 deben soportar IPSec, por lo que la protección IPv6 es igual de fuerte como IPv4 con IPSec. Sin embargo, a causa de la eliminación

Cantidad de bits:

4	4	8	16	
versión	Longitud de cabe-cera	Tipo de servicio	Longitud completa	
identificación			Bits de senal (3 bity) Don't Fragment, More Fragments	Número de fragmento
Actividad economica		Número de protocolo	Cabeceras aseguradas	
Direccion de fuente IP				
Direccion de fuente IP				
Posiblidad de eleccion				
datos (maximal (65535-longitud de cabecera) de octet)				

Figura 8. *Formato del datagrama IPv4*



cantidad de los bits:

4	4	8	8	8
versión	prioridad	Determinación de flujo de los datos		
Longitud de los datos en datagrama			Siguiente cabecera	Cantidad maxima de saltos
Direccion de fuente Ipv6				
Direccion final Ipv6				

Figura 9. *Formato de la cabecera obligatoria IPv6 del datagrama*

de NAT la implementación de IPSec, sobre todo, en las amplias redes es bastante más fácil.

Métodos de conectar las redes de IPv4 e IPv6

Introducir IPv6 en una red significa no solamente el cambio de protocolo de red clave y direccionamiento, sino que también los siguientes protocolos relacionados: ICMP, protocolos de routers y de aplicaciones. ICMPv6 incluye también funciones de los protocolos ARP (*Address Resolution Protocol*) y IGMP (*Internet Group Management Protocol*) con la versión 4. El protocolo RARP se excluirá completamente de la arquitectura de protocolo de la futura generación. Los protocolos de transporte TCP y UDP que trabajan sobre IPv6 no se diferencian mucho de los protocolos existentes de transporte para IPv4.

Entre los protocolos de aplicaciones (administrativos), el soporte para IPv6 es la extensión de los protocolos DNS y FTP ya existentes. Los protocolos de aplicaciones de usuario, tales como HTTP y SMTP no cambian. El paso de IPv4 a IPv6 se realiza por medio de los siguientes mecanismos básicos (se pueden combinar):

- *túnel* (RFC 3056; ver Figura 13.) – no requiere cambio en las estaciones finales y routers, conecta las nubes aisladas de IPv6 con el túnel por medio de la red con IPv4 (hermetización para los datagramas IPv4) entre los routers extremos duales IPv4/IPv6 (RFC 2893),
- *implementaciones duales de IPv4 en vez de IPv6* (*dual-stack*; ver Figura 14.) – implementaciones duales de los dos protocolos en los dispositivos en red,
- *traducción* – posible comunicación entre el dispositivo IPv4 y el dispositivo IPv6.

Los mecanismos provisionales abren las nuevas posibilidades de ataque: sobre todo los túneles creados de manera automática son susceptibles a la modificación de los paquetes y a los ataques de tipo DoS, por lo tanto es necesario protegerlos contra el empleo de IPSec. Se trata de un carácter de ataques parecido al de los túneles empleados en IPv4 – se diferencian solamente por el método de realización. Los túneles estáticos son más seguros pero sus capacidades de escalar no son tan altas. Los túneles pueden además evitar los servicios de tipo cortafuegos; también deberían localizarse solamente entre los sistemas autorizados. En cuanto a la doble implementación (*dual stack*) es necesario cuidar igual de bien de las protecciones de ambos protocolos. Los ataques contra IPv6 pueden fácilmente violar la red IPv4.

Posición de IPv6 en las redes modernas

Japón y China pertenecen a los países donde IPv6 tiene preferencia no solamente en las redes que se construyen nuevamente sino también en las redes de la administración pública. En Asia muchas veces encontramos el miedo al insuficiente espacio de direcciones que es limitación importante para el tiempo de desarrollo de la comunicación en red y tecnología. En Japón IPv6 pronto será el protocolo de comunicación en las redes públicas y comerciales. En China el año pasado se ejecutó la mayor red de investigación basada en IPv6, así denominado CERNET2 (*http://www.edu.cn*) que incluye a unas doscientas universidades y que está construida precisamente con IPv6.

Tanto en Europa como en EE.UU., el empleo de IPv6 se limita solamente a las redes de investigación y académicas, o bien para proyectos futuros. En EE.UU. existen, sin embargo, planes de paso de las organizaciones gubernamentales a IPv6 para el año 2008 como tarde. Los proveedores de Internet empiezan a conocer IPv6, sin embargo, teniendo en cuenta el poco soporte de los proveedores de las redes que se basan solamente en IPv6, los usuarios pueden hoy en día utilizar el modo de túnel de IPv6 por las redes IPv4. En cuanto al empleo de los túneles hay un problema, pero el entorno

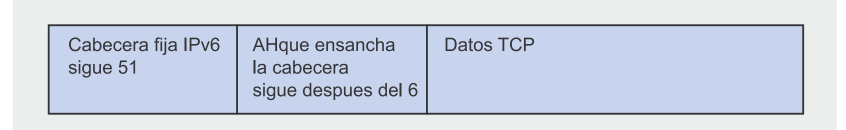


Figura 10. *Ejemplo que extiende las cabeceras del datagrama IPv6 para los datos TCP*

IPv6 es prácticamente accesible para todos los partidarios en todo el mundo. Sin embargo, la conectividad final puramente a base de IPv6 es solamente cuestión de tiempo.

Los implantadores europeos no se atrasan en cuanto a los requisitos relacionados con los bloques de direcciones IPv6 desde RIPE NCC, sin embargo, se atrasan, sobre todo, a causa de los requisitos de inversión en cada dispositivo de red y del soporte de IPv6 en cuanto a los servicios y aplicaciones en sus redes. Una serie de aplicaciones emplean actualmente las direcciones IPv4 en vez de los nombres (hostname), por lo tanto, el paso a IPv6 requiere cambios por parte de los clientes y servidores.

La primera red de investigación para soportar IPv6 fue 6Bone, creada ya en el año 1995. A los proyectos que propagaron más IPv6, perteneció Euro6IX (*http://www.euro6ix.org*). En su base se construyó la primera no comercial red IPv6 Internet Exchange y 6NET (*http://www.6net.org*). Esto confirmó la necesidad de pasar a IPv6 para el futuro desarrollo de Internet. En la construcción de la red 6NET participaron treinta socios, incluyendo el consorcio checo CESNET que ya a principios del año 2003 ejecutó en su país – dentro de 6NET – el primer circuito internacional realizado a base del protocolo IPv6 (no se realizó con el método de túnel).

La auténtica red CESNET2 creada en el protocolo IPv6 fue iniciada ya en el año 1999 y a partir del año 2004 el protocolo IPv6 es ofrecido como servicio explotado de manera estándar. En vez de un ordinario túnel de IPv6 sobre IPv4 CESNET2 implementó un mecanismo de la transmisión de datagramas IPv6 por medio de MPLS (*MultiProtocol Label Switching*) a base de la tecnología comercial 6PE (soportada por Cisco Systems y Juniper) y los datagramas IPv4 e IPv6 se transportan en una red real de manera totalmente igual. Algunas redes finales conectadas a CESNET2 también soportan IPv6 además de IPv4 en el modo *dual stack*, por lo tanto, la conexión a la red real MPLS por medio del router fronterizo por parte del cliente es más fácil. En el resto de los casos es necesaria la implementación de los distintos routers para soportar IPv6; debe también emplearse el mecanismo según IEEE 802.1Q para separar la explotación de IPv6 de IPv4.

La red nacional para investigación y educación (NREN) CESNET2 es también un componente de la red europea IPv6 GÉANT, donde tuvo lugar la explotación rutinaria de IPv6 iniciada en el año 2003. Su sucesor, GÉANT2 (*http://www.geant2.net*), una red construida a partir de cero, paneuropea de muchos gigabytes está destinada para los objetivos de investigación y académicos, soporta IPv6 y IPv4 de igual modo *dual stack,* ya descrito. GÉANT2 cuya primera fase ya ha finalizado, servirá para analizar la tecnología de red y aplicaciones incluyendo IPv6.

Existe una organización pública internacional que monitoriza y soporta la aplicación IPv6 en diferentes países y en diferentes tipos de redes. Se llama IPv6 Forum. Trabaja en la aplicación logo para la aplicación IPv6 Ready! (*http://www.ipv6ready.org*). Forum publicó, entre otros, el análisis en el cual advierte contra el potencial empleo de las direcciones IPv4 ya sobre el año 2008 (más información: *Internet Protocol Journal 9/2005*).

En las redes de investigación y académicas IPv6 está sometido a los permanentes y animados análisis, sin embargo, en las demás redes el empleo de estas redes se realiza más bien de forma obligatoria, ya que hay pocas personas que estén interesadas de este protocolo. Existe un verdadero precipicio entre los diseñadores e implementadores de los cuales verdaderamente depende la popularización del protocolo y su empleo de la práctica. Mientras que el primer grupo declara la necesidad de existir y emplear IPv6, el segundo lo ve de forma diferente: *IPv4 sigue sirviendo muy bien y, por el momento, no faltan direcciones.* Según las investigaciones realizadas por Juniper Networks en el grupo de directores IT y empleados de la administración gubernamental de EE.UU. solamente un 7% de los 349 preguntados considera el protocolo IPv6 importante para sus objetivos, aunque en el interés de los directores IT está tanto la fácil administración de redes como la mejora de la calidad de comunicación y, desde luego, el cumplimiento de los requisitos en cuanto a las

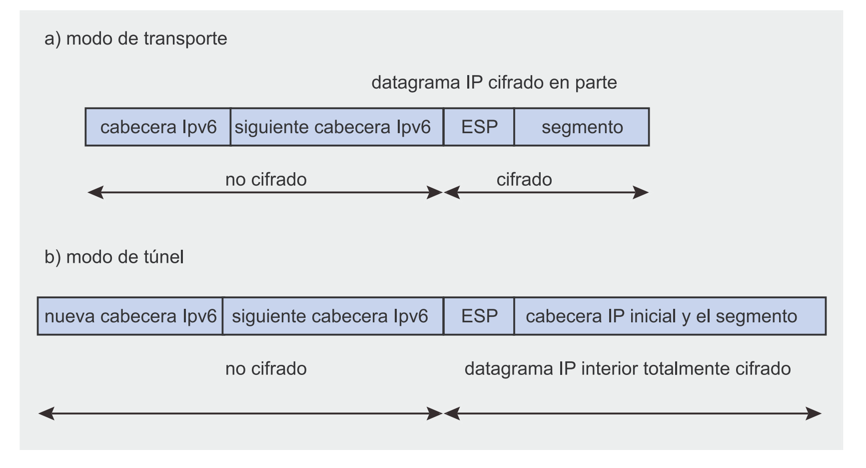


Figura 11. *Cifrado de los datos en el datagrama IPv6*

Tabla 3. *Diferencias en los servicios IPv4 e IPv6*

Servicio	solución IP versión 4	solución IP versión 6
direccionamiento	direccionamiento de 32-bits: individual (unicast), en grupos (multicast), general (broadcast)	direccionamiento de128-bits: individual (unicast), en grupos (multicast), frontera en el grupo (anycast)
autoconfiguración	posibilidad de DHCP para los nudos finales	DHCPv6 (stateful, servidor necesario); auto-configuración (stateless, sin la necesidad del servidor), cambio de numeración de routers (router renumbering)
calidad de servicios (QoS)	bits IP precedencia (en el campo Type of Service), cambio en bits para DiffServ; siguiente mecanismo IntServ	Clases de explotación, identificación de los respectivos flujos de datos (priority & flow) directamente en el datagrama
capacidad de escalar	routing con CIDR, agregado de direcciones para prefijos (bloques de direcciones)	routing jerárquico a base de la estructura jerárquica de las globales
emisión en grupo	direccionamiento en grupo (multicast) – para la emisión a todos los miembros del grupo dado (por ejemplo, video conferen-cias, formación en línea)	direccionamiento en grupos para todo el grupo de estaciones (multicast) direccionamiento de una (la más próxima) frontera en grupo (anycast) – para la distribu-ción eficaz de los requisitos de los usuarios
protección	IPSec – posible completación	soporte obligatorio; parte del datagrama: cabecera de extensión (AH, ESP)
movilidad	Mobile IP – routing por un agente de casa en el agente ajeno, donde se encuentra conectado el usuario (triangle routing)	eliminación de túnel por el agente de casa, routing directo

protecciones. Además, el protocolo de la nueva generación tiene que reducir el coste de la transmisión en grupo (*multicast*) de vídeo por medio de IP y simplificar el paso a VoIP. Lo más importante es que – para una tercera parte de los preguntados por Juniper Networks la causa del seguimiento con IPv4 es... falta de la motivación para cambios. La siguiente, tercera parte quisiera cambio, sin embargo – se-gún la opinión de los investigados – les molestan los gastos relacio-nados con tal cambio (sobre todo la necesidad de cambio del antiguo hardware). Apenas un 17% teme problemas técnicos.

La mayoría de las aplicaciones IPv6 sigue encontrando el obstá-culo de gastos que actualmente no puede permitirse una serie de usuarios, sobre todo cuando no son capaces de determinar los provechos en forma de ahorros, adelanto de la competencia etc. Sigue dominando la opinión según las cual falta una aplicación infor-mática determinante (*killer appli-cation*), para la cual el paso a IPv6

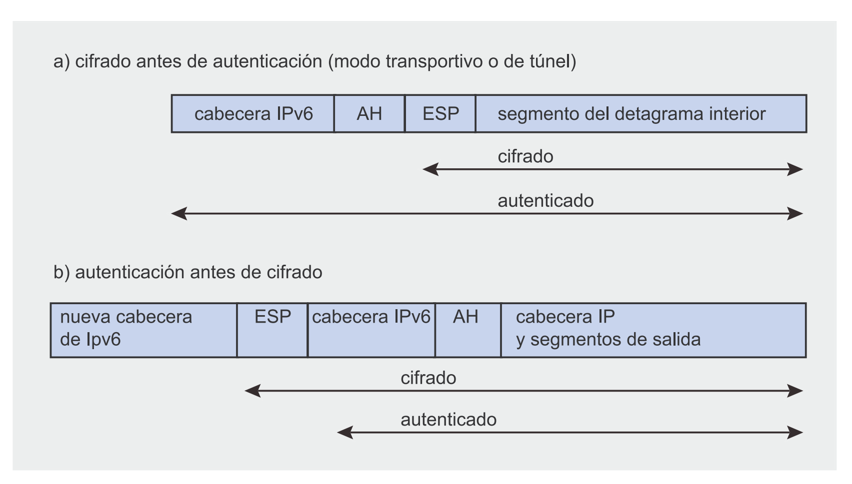


Figura 12. *Cifrado y autenticación en IPv6*

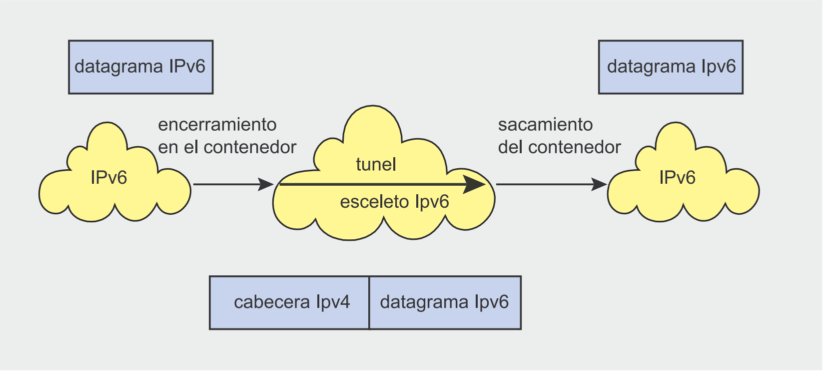


Figura 13. *Túnel IPv6 por IPv4*

Tabla 4. *Diferencias en la cabecera del datagrama IPv4 e IPv6*

Información en el datagrama	IP versión 4	IP versión 6
largo de la cabecera	variable – necesidad de especificar en el campo largo de la cabecera	constante – sin necesidad de especificar en el datagrama
largo del datagrama	la especifica el valor en la longitud general	No se da, solamente la longitud de una parte de los datos – el campo payload length
tipo de servicio	campo ToS – hoy para DSCP	campo de prioridad y *flow label*
duración	etiqueta en el campo TTL	señalada en el campo *hop limit*
fragmentación	campo de identificación, aviso, offset para fragmentar el datagrama en el camino	fragmentación solamente en la fuente – solamente la cabecera extendida para fragmentar
protocolo superior	campo número de protocolo	reemplazada con el campo la siguiente cabecera
suma de control de cabecera	campo protección	no se realiza, se cuenta con protocolos superiores
posibilidades seleccionables	Campo que ayuda al variable largo del datagrama	No son componente de la cabecera constante pero extendida

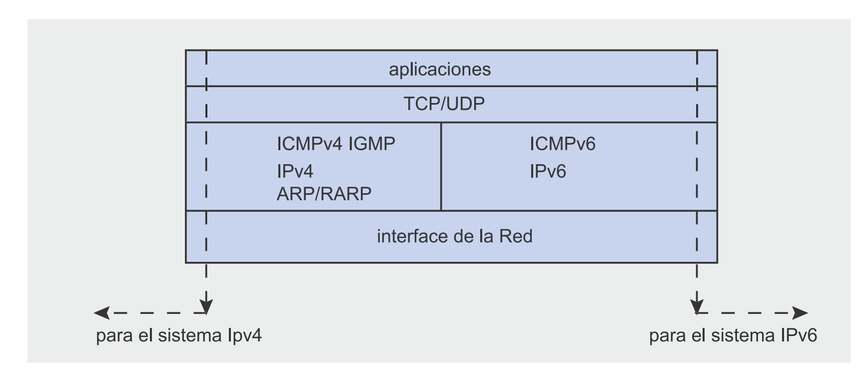


Figura 14. *Dual stack*

merecería la pena. La verdad es que la suficiente *justificación* para el paso al nuevo protocolo constituye un gigantesco y creciente todo el tiempo número de los usuarios de teléfonos móviles o bien cada vez más grande popularidad de las redes de ordenadores de casas. Además, podemos hablar de un futuro luminoso de así denominada electrónica de consumo, de un creciente interés de minisensores ye identificaciones (RFID)...

Intervalos problemáticos

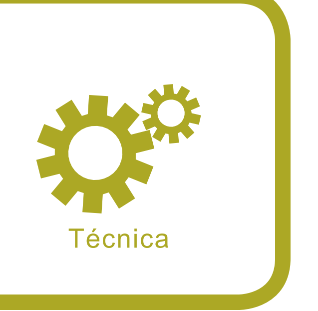
IPv6 es un cómodo protocolo de futuro, sin embargo, su protección todavía no es completa. La política de protecciones, los procesos y los medios deben ajustarse a IPv6. Es necesario recordar la instalación de cortafuegos con el respectivo soporte – suele ser un problema, ya que los productos presentes en el mercado casi nunca tienen en cuenta las necesidades típicas para IPv6. Una situación parecida se refiere a la estabilidad y la complejidad del soporte de todos los elementos de IPv6 en los routers y sistemas operativos.

Un firewall en las redes IPv6 cambia su carácter, teniendo en cuenta la modificación en el modelo general de red. Tiene lugar el cambio de su localización en la red o bien en el sistema final (firewall personal) con lo cual – para cambiar la administración (administrador o usuario), ya que no existe una clara frontera de redes que deben protegerse. De los sistemas finales se espera más protección: con contraseña protegida, protección antivirus, cifrado de datos etc.

Con respecto al paso de IPv4 a IPv6 hay que dedicar suficiente atención ya que la migración no se realizará inmediatamente – ambos entornos coexistirán mucho tiempo. Son justamente los mecanismos provisionales – túneles, doble implementación en los dispositivos – pueden ser problemáticos en el contexto de protecciones. Además de los susodichos obstáculos, la protección de los sistemas finales, la protección física, la protección de aplicaciones y el control de protecciones seguirán siendo una cuestión muy importante de las cual decidirán los administradores de redes y directores de protecciones. ●

En la Red

- *http://www.ins.com/downloads/surveys/sv_ipv6_1205.pdf* – INS IT Industry Survey: Ipv6
- *http://hexateuch.6net.org/thebook/* – Ipv6Deployment Guide
- *http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html* – A Pragmatic Report on IPv4
- *http://www.SANS.org* – SANS Institute
- *http://www.ietf.org/rfc.html* – RFC



Ingeniería Inversa: Desensambladores de tamaño

Rubén Santamarta 

Grado de dificultad



Día tras días los investigadores de malware, analistas forenses o administradores se tienen que enfrentar con amenazas a la seguridad en los sistemas de información. El objetivo puede ser esclarecer instrusiones no autorizadas, proteger a los usuarios de virus o evitar que un sistema sea comprometido.

Los creadores de malware (virus, troyanos, rootkits) intentan dificultar lo máximo posible éste análisis, utilizando para ello técnicas antidebug, polimorfismo, stealth o packers entre otras; estos últimos a la vez que reducen el tamaño del ejecutable, añaden con mayor o menos complejidad una capa adicional de protección.

En estas situaciones el tiempo empleado es una pieza clave, no hay duda que con un análisis pausado y exhaustivo más tarde o más temprano conseguiríamos nuestro objetivo de conocer todos los detalles de la amenaza. Desafortunadamente hay ocasiones en las que no se dispone de todo el tiempo que sería deseable y hay que optimizar todas las acciones destinadas al análisis. Imaginemos un gusano explotando una vulnerabilidad no conocida para propagarse a través de Internet; el tiempo invertido en analizar y comprender su funcionamiento, marcará la diferencia entre una aunténtica catástrofe para los usuarios y una amenaza neutralizada y reducida.

Debemos por lo tanto, dotarnos con los recursos suficientes para solventar cualquier tipo de dificultad que se nos presente.

Hooking

Como hemos visto, existen multitud de trucos destinados a dificultarnos la utilización del debugger (tanto en Ring0 como Ring3), herramienta básica en la ingeniería inversa. Por este motivo, debemos construir un método que, bajo determinadas circunstancias, nos permita interactuar y modificar el comportamiento del ejecutable sobre el que estamos investigando.

Una de las técnicas más usadas para conseguir este objetivo, es el hooking.

En este artículo aprenderás...

- Cómo aplicar el hooking en el análisis de malware
- Cómo usar Structure Exception Handling para crear un desensamblador de tamaño.

Lo que deberías saber...

- Ensamblador x86 y C
- Conocimiento de Win32 Api y Structure Exception Handling
- Conocimientos básicos de técnicas usadas por malware y virus.

Técnicas contra desensambladores y debuggers

A lo largo de los años los creadores de malware, los escritores de virus o incluso los propios programadores de software comercial han dotado a sus creaciones de métodos antidebug y antidesensamblado. La mayoría de ellos están destinados a detectar si el programa está siendo observado por un debugger. En caso afirmativo las acciones tomadas por el programa pueden ser variadas, desde terminar su ejecución drásticamente hasta reiniciar el ordenador o incluso algunas más agresivas pero, afortunadamente, mucho menos comunes.

- Un viejo truco usado (todavía usado) para detectar la presencia del SoftIce, el debugger de Ring0 más conocido y usado en el mundo de la ingeniería inversa, consistía en intentar acceder a los dispositivos creados por uno de sus drivers, NtIce.
- La instrucción en ensamblador x86 `RDTSC`:*mnémonico de Read Time−Stamp Counter*. Esta instrucción guarda en `EDX:EAX` (64 bits) el valor del *timestamp* del procesador. Imaginemos que `RDTSC` es ejecutada al comienzo de un bloque de código y el valor devuelto almacenado. Al final de ese bloque de código volvemos a ejecutar `RDTSC` y restamos el valor obtenido con el anteriormente almacenado. Bajo condiciones normales de ejecución, el resultado de este operación se puede acotar entre unos valores razonables,la velocidad y la carga del procesador obviamente influirán, pero si estamos depurando ese bloque de código, el incremento del *timestamp* entre ambas lecturas se disparará, con lo que habremos descubierto al debugger.
- Manejo de interrupciones para cambiar el flujo del código. Una potente carácteristica de la arquitectura Win32 es el *Structure Exception Handling* (SEH), el cual nos permite establecer rutinas de *callback* para controlar excepciones. Debido a que los debuggers suelen ocuparse de cualquier excepción ocurrida durante la ejecución del programa, el procedimiento que el programador haya establecido para manejar excepciones no llegará nunca a ejecutarse. Supongamos que hemos basado el flujo de nuestro programa en un procedimiento de este estilo, si al provocar deliberadamente una excepción (usando por ejemplo `xor eax,eax` y después `mov [eax],eax`), no alcanzamos la zona de código prevista, probablemente estemos bajo la supervisión de un debugger.
- Otros trucos menos elaborados se basan en características inherentes a cada debugger. Ya sea intentando encontrar determinadas clases o títulos de ventanas registradas por el programa, o símplemente buscando claves en el registro de *Windows* que puedan delatarle.

Figura 1. *Esquema básico del Inline Hooking*

Podríamos clasificar brévemente las distintas técnicas de hooking atendiento al lugar donde se produce. Cada tipo está orientado hacia distintas aplicaciones. Así tendríamos los siguientes tipos:

- Inline Hooking
- Import Address Table hooking
- System Service Table hooking (Ring0)
- Interrupt Descriptor Table hooking (Ring0)
- IRP hooking (Ring0)
- Filter drivers (NDIS, IFS...Ring0)

El método que aplicaremos será el Inline Hooking. La razón es que usando esta técnica, lo que hacemos es parchear directamente la función que nos interesa interceptar cuando esta cargada en memoria. De esta manera, no nos preocupamos desde qué lugar se la está referenciando, o cuantas veces. Atacamos directamente a la raiz. Cualquier llamada a la función, sera interceptada por nuestro gancho.

Interceptando y modificando el flujo del código

Supongamos que deseamos interceptar todas las llamadas a la API CloseHandle que se producen durante la ejecución de un programa. Dicha API se encuentra localizada en kernel32.dll, veamos cómo son sus primeras instrucciones:

```
8BFF    mov   edi,edi
55   push   ebp
8BEC   mov   ebp,esp
64A118000000  mov
   eax,fs:[00000018]
8B4830  mov   ecx,[eax][30]
8B4508  mov   eax,[ebp][08]
```

Este bloque de código representa los primeros bytes del punto de entrada de *CloseHandle*, es decir, cualquier llamada a dicha función ejecutará inevitablemente el código anterior. Observando el esquema del *Inline Hooking*, estas primeras instrucciones serán sobreescritas por nuestro propio gancho, el cual modificará el flujo normal de la función hacia el filtro.

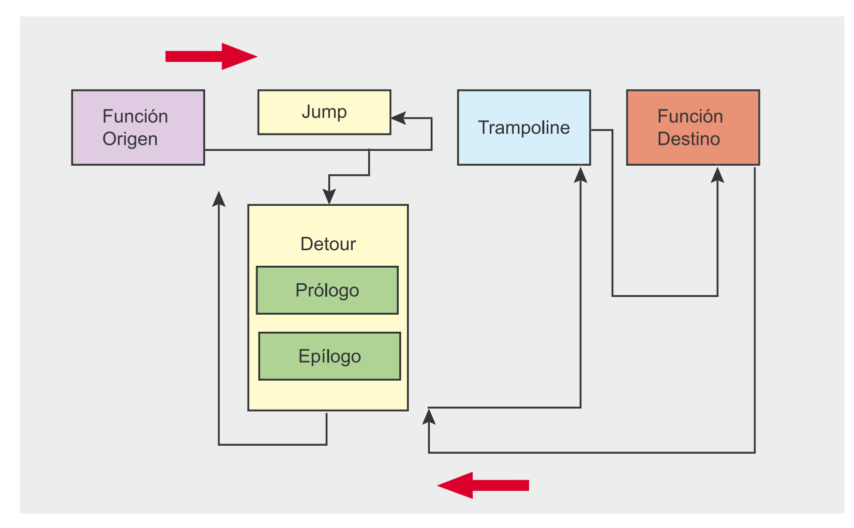


Figura 2. *Esquema básico de la técnica Detour*

Tabla 1. *Datos accesibles por el manejador de excepción cuando es activado*

En	Dato
ESP+4	Puntero a la estructura EXCEPTION_RECORD
ESP+8	Puntero a la estructura ERR
ESP+C	Puntero a la estructura CONTEXT_RECORD

Tabla 2. *Campos de la estructura de EXCEPTION_RECORD*

Offset	Dato
+ 0	ExceptionCode
+ 4	ExceptionFlag
+ 8	NestedExceptionRecord
+ C	ExceptionAddress
+ 10	NumberParameters
+ 14	AdditionalData

Tabla 3. *Campos de CONTEXT pertenecientes a los registros generales y de control*

Offset	Registro
+ 9C	EDI
+ A0	ESI
+ A4	EBX
+ A8	EDX
+ AC	ECX
+ B0	EAX
+ B4	EBP
+ B8	EIP
+ BC	CS
+ C0	EFLAGS
+ C4	ESP
+ C8	SS

Códigos de excepción

No se puede tratar de la misma manera una excepción producida por un acceso a una posición de memoria no válida, que a una excepción producida por una división por 0. Debido a esto, el sistema identifica unívocamente cada situación para facilitar al manejador de excepciones su labor. Algunos de los códigos de excepción más habituales son los siguientes:

- *C0000005h* – Violación de acceso en operaciones de lectura o escritura,
- *C0000017h* – No existe memoria disponible,
- *C00000FDh – Stack Overflow.*

Las dos siguientes son de vital importancia para nuestro proyecto:

- *80000003h* – Breakpoint generado por la instrucción `int 3`,
- *80000004h* – Single step generado por la activación del *Trap Flag* en el registro *EFLAGS*.

Diferentes posibilidades para un mismo propósito

El método para desviar el flujo hacia nuestro código puede variar. El más sencillo de todos sería sobreescribir los primeros bytes de CloseHandle con un salto incondicional.

```
01  E9732FADDE   jmp   0DEADBEEF
02  64A118000000  mov   eax,fs:
[00000018]
```

Hemos sobreescrito los primeros 5 bytes con un salto hacia la dirección 0xDEADBEEF, obviamente esta dirección no es válida ya que estamos trabajando sobre modo usuario. En esta dirección se encontraría el código que hayamos inyectado en el espacio de direcciones del ejecutable.

Al ser el método más habitual también es el más detectable debido a que resulta extremadamente sospechoso que el *entry point* de una función del sistema contenga un salto incodicional hacia otra dirección de memoria. Podemos usar

Listado 1. *Puntos plasmados en código ensamblador*

```
SEH_SEHUK:
mov   esi, [esp + 4]     ; EXCEPTION_RECORD
mov   eax, [esi]          ; ExceptionCode
test   al,  03h              ; Int3 Exception Code
mov   eax, [esi + 0Ch] ; Eip Exception
mov   esi, [esp + 0Ch] ; CONTEXT record
mov   edx, [esi + 0C4h]   ; Esp Exception
jz    Int1h
mov   eax, [esi + 0B4h]   ; Ebp Exception
mov   [OrigEbp], eax
mov   [OrigEsp], edx
inc    dword [esi + 0B8h]  ; Eip++ (Int3->Instrucción siguiente)
mov   eax,Code
mov   [PrevEip],eax
jmp   RetSEH
```

esta otra opción; la combinación de PUSH + RET.

```
68EFBEADDE  push  0DEADBEEF
C3        retn
A118000000  mov  eax,[00000018]
```

En esta ocasión sobreescribimos los primeros 6 bytes. Si observamos atentamente nos daremos cuenta que el código original de *CloseHandle* ha variado sustancialmente, pero no sólo por las instrucciones añadidas, sino que algunas de las ya existentes se han perdido debido a que las hemos sobreescrito y las siguientes han pasado a ser totalmente diferentes. Esto se presenta como un problema a considerar seriamente ya que aunque si bien es cierto, que hemos cumplido nuestro objetivo de interceptar todas las llamadas a la función, tambiés es cierto que la modificación de su código original ha sido los suficientemente sustancial como para para provocar un comportamiento anómalo, dando como resultado que, con toda seguridad, el programa termine inesperadamente a la primera llamada a *CloseHandle*.

Se hace necesario entonces desarrollar una técnica lo menos agresiva posible contra el código original, que permita a la función hookeada seguir comportándose como si nada estuviera pasando, pero que a su vez la podamos seguir controlando. Esta técnica se conoce como *Detour* (presentada por *Galen Hunt* y *Doug Brubacher* de los laboratorios *Microsoft*).

Detour

Respecto al Inline Hooking, la técnica *Detour* introduce dos conceptos nuevos como la Función Detour y la función Trampoline.

- La Función *Detour* debería constar de una primera parte donde se realizarían las primeras operaciones sobre los datos recibidos, posteriormente la llamada a la función Trampoline y por último, una porción de código que se ejecutará cuando se haya completado la función *Trampoline*.
- La Función *Trampoline* contiene tanto las instrucciones de la función destino, sobreescritas completamente por el salto incondicional (JUMP), como las que hayan sido modificadas parcialmente. A continuación tendremos un salto hacia la siguiente instrucción que corresponda en la función destino.

De esta manera hemos solucionado el problema de las instrucciones perdidas o modificadas que teníamos con el Inline Hooking. La clave está en salvar esas instrucciones en la Función Trampoline para que sean ejecutadas. Posteriormente, saltaremos hacia la siguiente instrucción donde la función destino seguirá normalmente. Una vez que la fun-

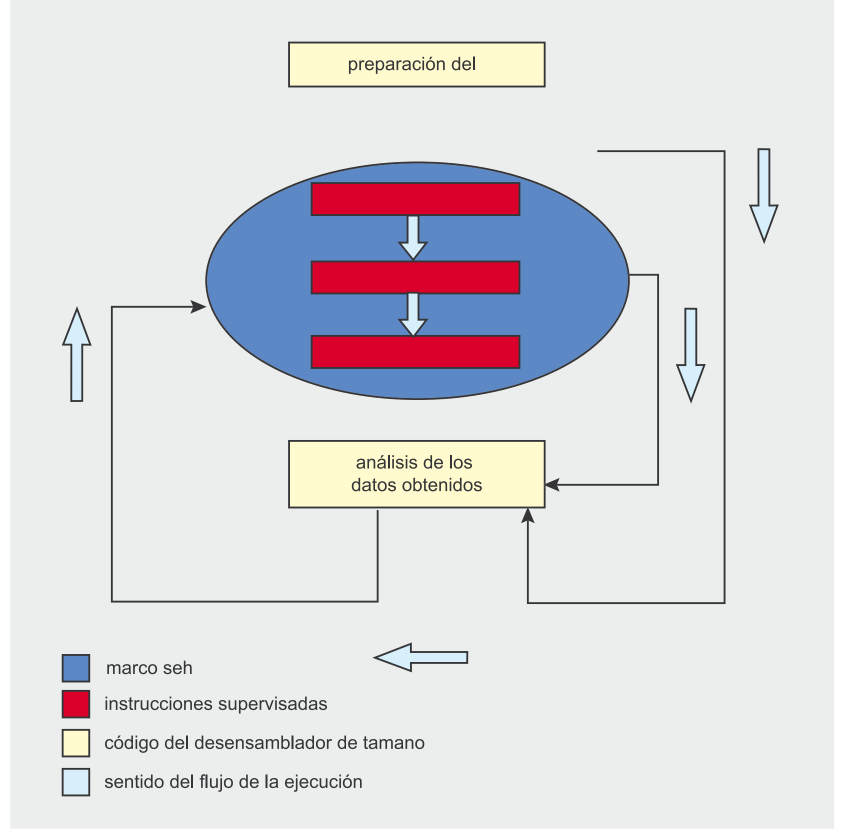


Figura 3. *Esquema de funcionamiento de nuestro desensamblador de tamaño*

La manera que de llamar al programa congrio c:\acpinject.exe 10000 Kernel32.dll ExitProcess

- como primer argumento tenemos el path del malware
- como segundo argumento el intervalo en milisegundos que pasaremos a Sleep
- el tercer argumento es la DLL que exporta la función destino
- La *función destino*, en este caso *ExitProcess* (ver Listado 4)

ción destino ha sido completada, volvemos a tener el control en el epílogo de la Función Detour. Ésta tiene la opción de restaurar el camino de la ejecución, devolviendo el control a la función origen o realizar otro tipo de operaciones.

Ahora bién, ¿cómo conocemos cuantas instrucciones deberemos copiar a la Función Trampoline desde la función destino? Cada función destino será diferente, por lo que no es posible copiar una cantidad fija de bytes ya que podríamos estar cortando instrucciones. Este problema se resuelve usando los desensambladores de tamaño.

Desensambladores de Tamaño

Los desensambladores de tamaño difieren de los desensambladores habituales en que su única misión es obtener la longitud de las instrucciones y no representarlas. Este tipo de desensambladores han sido usados tradicionalmente por virus cavity, polimórficos, etc. De ahí que dos de los desensambladores de tamaño (auténticas joyas de la optimización extrema) más usados, hayan sido programados por conocidos escritores de virus: Zombie y RGB.

Estos se basan en un desensamblado estático de las instrucciones.

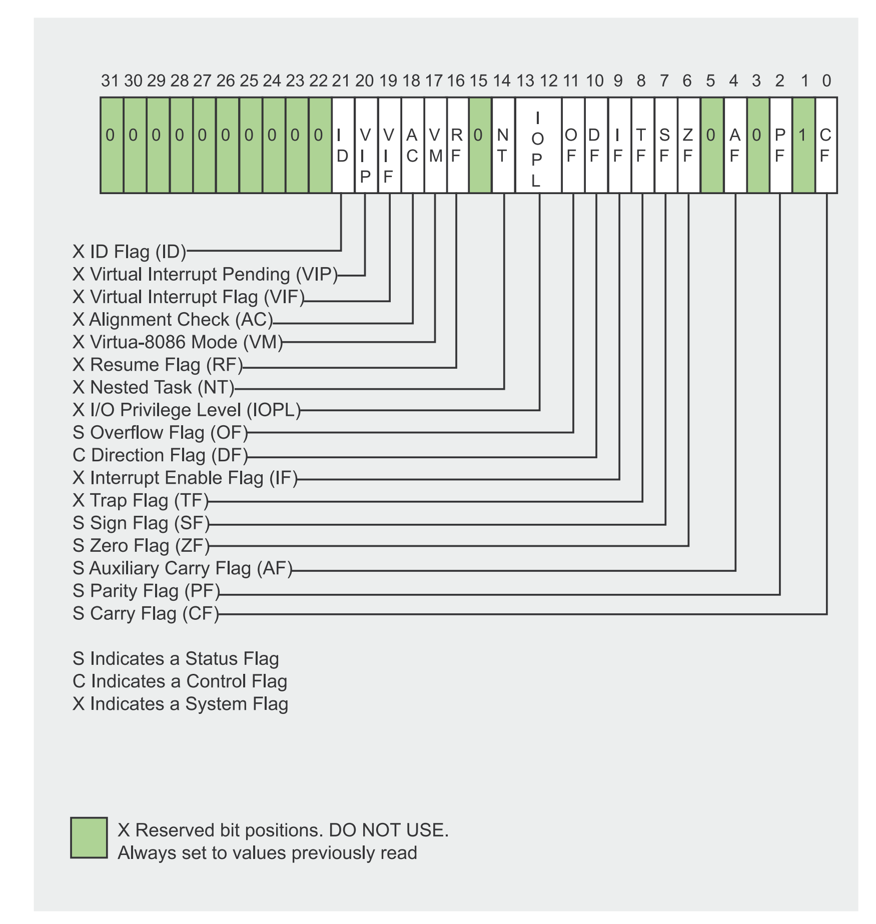


Figura 4. *Registro EFLAGS*

Listado 2. *Código perteneciente al análisis de las instrucciones supervisadas*

```
Int1h:
mov   ecx, eax
sub    eax, [PrevEip]
cmp   ax, 10h
jb      NoCall
mov  ebx, dword [edx]
mov  edx,ebx
sub   ebx, [PrevEip]
cmp bl, 7
jbe   HabemusCall
mov   edi,[PrevEip]
inc     edi
inc     edi
mov     dword
 [esi + 0B8h], edi
mov    ecx,edi
jmp     NoCall
HabemusCall:
mov   dword
 [esi + 0B8h], edx
mov   ecx,edx
NoCall:
mov    [PrevEip], ecx
sub   ecx, Code
cmp   ecx, [HookLength]
jge     Success
RetSEH:
or  word [esi + 0C0h], 0100h  ;
 Activamos  Trap Flag
xor    eax,eax
ret
Success:
mov     [LenDasm], ecx;
Devolvemos
la longitud de las instrucciones
mov   esp, [OrigEsp];
Restauramos Esp
mov   ebp, [OrigEbp] ;
Restauramos Ebp
pop   dword [fs:0];
Limpiamos el marco SEH
add   esp, 4;
Ajustamo las pila
```

Usan para ello tablas de opcodes de la arquitectura sobre la que operan, en este caso x86.

A parte de su uso para la creación de virus complejos, son utilizados también para el hooking, debido a que con ellos se resuelve el problema anteriormente comentado.

Por ello, basándonos en la potencia del *Structure Exception Handling,* se explicará una técnica innovadora para crear un desensamblador de tamaño dinámico. Manos a la obra.

Aplicando el Structure Exception Handling (SEH)

¿Qué información podemos obtener a través del SEH? En primer lugar es conveniente fijarnos en las características del problema sobre el que queremos plantear la solución.

- Las primeras instrucciones de las funciones destino, no suelen variar demasiado, pero lo hacen lo suficiente como para que sea indispensable ajustarnos a cada caso individualmente.
- Un salto incondicional (jmp) o un `Push` + `ret` no va a ocupar más de 6 bytes. No será necesario analizar más de 4 ó 5 instrucciones como máximo.
- Las primeras instrucciones suelen realizar operaciones relacionadas con ajustes de la pila.

La idea es conseguir ejecutar estas primeras instrucciones en un entorno no controlado, lo que nos permitirá calcular su longitud.

Para intuir cómo podemos construir este entorno, debemos introducir la información que nos brinda SEH.

Para cada excepción ocurrida dentro del código protegido por un marco SEH definido para una thread, el manejador asignado tiene a su disposición los siguientes datos.

Una vez que se ha producido la excepción, el sistema activa el manejador de excepción para que éste determine qué hacer con ella. En ese momento `esp` está apuntando a diversas estructuras.

De la estructura EXCEPTION_RECORD prestamos atención a los campos *ExceptionCode* y *ExceptionAddress:*

- *ExceptionCode* es el identificador del tipo de excepción que se ha producido. El sistema tiene asignados diferentes códigos para cada tipo, además es posible definir nuestros propios códigos para personalizar una excepción a través de la API RaiseException.
- *ExceptionAddress* es la dirección de memoria perteneciente a la instrucción que ha generado la excepción, equivaldría al registro eip en el momento de producirse ésta.

La otra estructura básica a conocer es CONTEXT. Dicha estructura contendrá los valores de todos los registros en el momento en que se produjo la excepción.

Debemos tener en mente que lo principal en todo momento, es poder controlar cada instrucción ejecutada como si estuvieramos haciéndolo paso a paso con un debugger. De hecho vamos a aplicar a nuestro desensamblador de tamaño, el funcionamiento base de un debugger.

Programando el desenseblador de tamaño

Lo primero es definir el entorno donde ejecutaremos las instrucciones supervisadas; llamaremos así a las instrucciones pertenecientes a la función destino de las cuales queremos conocer su longitud, para poder posteriormente copiarlas íntegras a la *Función Trampoline*.

Preparación del entorno

Lo primero es definir un marco SEH donde SEH_SEHUK será la rutina que manejará las excepciones que ocurran.

```
push    dword SEH_SEHUK
push    dword    [fs:0]
mov    [fs:0], esp
```

A partir de ahora, todo el código que se ejecute después de estas instruccionies estará protegido. El siguiente paso es copiar una determinada cantidad de bytes,los cuales contendrán las *instrucciones supervisadas*, desde la función destino a un area reservada dentro de nuestro código. El tamaño de

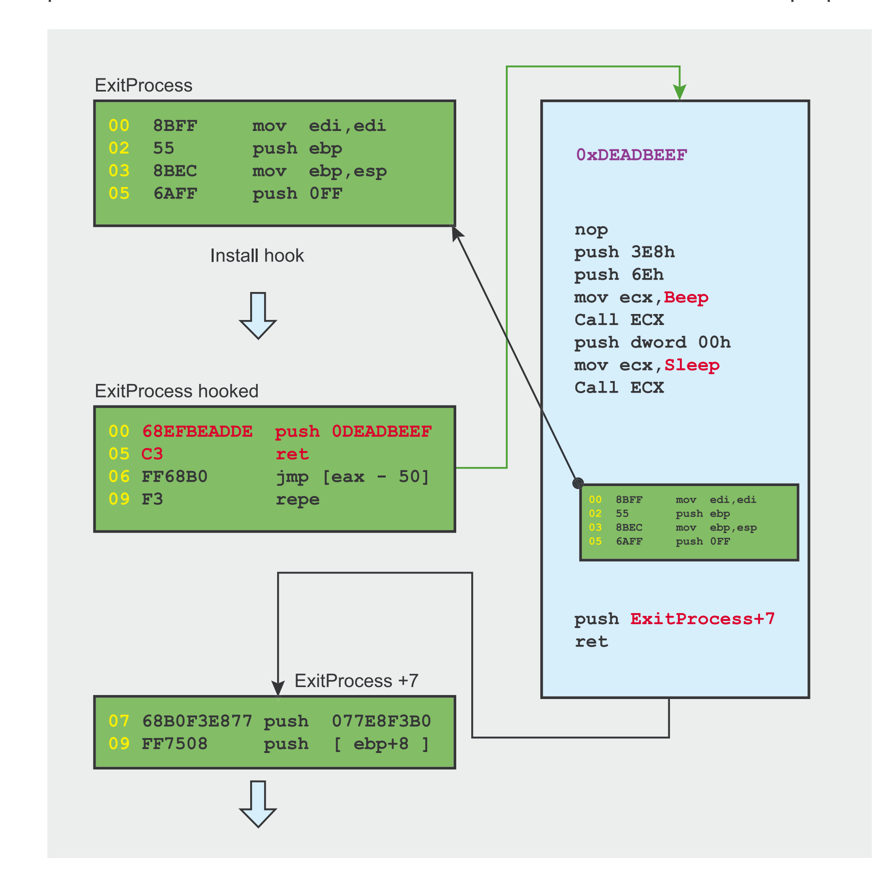


Figura 5. *Esquema de cómo se hookeará ExitProcess*

Listado 3. *ACPInject*

```
#include <stdio.h>
#include <windows.h>
typedef BOOL (WINAPI *PQUEUEAPC)(FARPROC,HANDLE,LPDWORD);
int main(int argc, char *argv[])
{
PROCESS_INFORMATION strProceso;
STARTUPINFOA        strStartupProceso;
PQUEUEAPC          QueueUserApc;
DWORD            MessageAddr,Ret1,Ret2,Longitud;
char              *szExecutableName;
unsigned char       Snippet[]=  "\x90"     /* nop     */
   "\x6A\x00"   /* push NULL  */
   "\x6A\x00"   /* push NULL  */
   "\x6A\x00"   /* push NULL  */
   "\x6A\x00"   /* push NULL  */
   "\xB9\x00\x00\x00\x00"  /* mov ecx,MessageBox*/
   "\xFF\xD1" ;   /* Call ecx   */

Longitud = (DWORD) strlen( "c:\\windows\\system32\\calc.exe") + 1;
ZeroMemory( &strStartupProceso, sizeof( strStartupProceso) );
strStartupProceso.cb = sizeof( strStartupProceso );
ZeroMemory( &strProceso, sizeof( strProceso) );
szExecutableName = (char*) malloc( sizeof(char) * Longitud);
if( szExecutableName )  strncpy(szExecutableName,
"c:\\windows\\system32\\calc.exe",Longitud);
else   exit(0);

_QueueUserApc = (PQUEUEAPC)GetProcAddress( GetModuleHandle (
"kernel32.dll" ), "QueueUserAPC");
MessageAddr = (DWORD) GetProcAddress ( LoadLibraryA(
"user32.dll") , "MessageBoxA" );

// U32!MessageBoxA
*( DWORD* )( Snippet + 10 ) = MessageAddr;
  Ret1 = CreateProcessA( szExecutableName,    NULL, NULL, NULL,
   0, CREATE_SUSPENDED,
   NULL,NULL,
   &strStartupProceso,&strProceso);

 Ret2 = (DWORD) VirtualAllocEx( strProceso.hProcess,NULL,sizeof(Snippet),
   MEM_COMMIT,
   PAGE_EXECUTE_READWRITE);

  WriteProcessMemory(strProceso.hProcess,(LPVOID)Ret2,  Snippet,
 sizeof(Snippet), NULL);
 _QueueUserApc((FARPROC)Ret2,strProceso.hThread,NULL);
  ResumeThread(strProceso.hThread);
 return 0;
}
```

éste puede variar. En este caso hemos elegido 010h por ser lo suficientemente amplio como para albergar las primeras instrucciones completas.

```
mov    esi,TargetFunction
mov    edi,Code
push    010h
pop     ecx
rep     movsb
```

Una vez hemos alcanzado este punto, sólo nos queda un paso antes de empezar a ejecutar las instrucciones supervisadas. Veámoslo primero.

```
int 3
Code:
ModCode      times 12h db (90h)
```

ModCode es el espacio donde hemos copiado nuestras instrucciones supervisadas, observamos que justo antes de llegar a este punto, hemos colocado una `int 3`, ¿por qué? Varios motivos nos obligan a ello:

- Como anteriormente mencionábamos, las primeras instrucciones de cualquier función destino, suelen realizar operaciones de modificación de la pila. Por este motivo, debemos asegurarnos que el estado de nuestra pila no se corrompe debido a esto. Al ejecutar int 3 generamos una excepción que nos servirá para entrar en *SEH_SEHUK*, nuestro manejador. De esta manera accediendo a la estructura *CONTEXT* salvaremos los registros *ESP* y *EBP* con el fin de una vez terminado nuestro análisis restaurar el estado de nuestra pila con los valores anteriores a que sufriera modificaciones.
- Activar el *Trap Flag* en el registro *EFLAGS*. Mediante esta técnica conseguimos que una

Listado 4. *Llamar al programa*

```
unsigned char HookCode[]=  "\x90"
     /* nop  */
  "\x68\x38\x03\x00\x00"
  /* push 3E8h  */
  "\x6A\x6E"  /* push 6Eh  */
  "\xB9\x00\x00\x00\x00"
  /* mov ecx,00h  */
  "\xFF\xD1"  /* Call K32!Beep */
  "\x68\x00\x00\x00\x00"
  /* push dword 00h    */
  "\xB9\x00\x00\x00\x00"
  /* mov ecx,00h  */
  "\xFF\xD1"  /* Call K32!Sleep */
  "\x90\x90\x90\x90"
/* Espacio para
las instrucciones supervisadas*/
  "\x90\x90\x90\x90"
  "\x90\x90\x90\x90"
  "\x90\x90\x90\x90"
  "\x90\x90\x90\x90"
  "\x68\x00\x00\x00\x00"
  /* push dword 00h    */
  "\xC3"  /* ret  */
  "\x90"; /* nop  */

unsigned char ExitHook[]=
  "\x68\x00\x00\x00\x00"
  /* push dword 00h    */
  "\xC3"; /* ret  */
```

Listado 5. *Construir HookCode*

```
/* Vamos construyendo nuestro
HookCode con las direcciones
obtenidas */
printf
("[+]Rebuilding HookCode...");

// K32!Beep
*( DWORD* ) ( HookCode + 9 ) =
 BeepAddr;

// Sleep param
Parameter = atoi( argv[2] );
*( DWORD* ) ( HookCode + 16 ) =
 Parameter;

// K32!Sleep
*( DWORD* ) ( HookCode + 21 ) =
 SleepAddr;

*( DWORD* ) ( HookCode + 48) =
 HookAddr + LenDasm;

printf("[OK]\n");
```

vez se ejecute la siguiente instrucción, automáticamente se generará una excepción *Single Step*. Con lo que volveremos a tener el control. Así hemos conseguido la misma información que obtendríamos depurando el programa paso a paso. (Vease Listado 1).

En la linea 15 estamos comparando con la 03 para tratar de averiguar si la excepción ha sido producida por una `int 3` (código de excepcion *80000003h*) o por el contrario se debe a una excepción producida por el *Trap Flag* o cualquier otro evento. En caso de que estemos ante una excepción de *BreakPoint* aplicaremos lo anteriormente explicado; líneas 20, 21, 22.

Es necesario modificar el *EIP* del contexto para que cuando devolvamos el control al sistema, apunte a la siguiente instrucción después de `int 3`, ya que si no entraríamos en un bucle sin salida. Para ello, como vemos en la línea 23, símplemente incrementamos en uno su valor. Esto es debido a que el opcode de `int 3` tiene el tamaño de 1 byte.

En la linea 25 guardamos la dirección de memoria donde empiezan nuestras *instrucciones* supervisadas, esto nos ayudará a la hora de emular llamadas y saltos condicionales o incondicionales.

Análisis de las instrucciones supervisadas

Hasta ahora, todo lo que hemos visto podríamos englobarlo dentro del bloque *preparación del entorno*. Empezaremos a ver el código perteneciente al análisis de las *instrucciones supervisadas (Vease Listado 2)*.

Objetivo

El objetivo de esta parte de código es analizar la longitud de las instrucciones supervisadas hasta encontrar un valor mayor o igual que el que ocuparía nuestro gancho, ya sea un salto incondicional (`jmp` 5 bytes) o un `push` + `ret` (6 bytes). Por ejemplo, imaginemos que nuestro gancho es del modo `push`+`ret` y estamos intentando hookear *CloseHandle*. Indicaremos a nuestro desensamblador de tamaño que nuestro gancho ocupa 6 bytes (HookLength = 6). Entonces empezaría a calcular la longitud de la primera instrucción:

```
01  8BFF    mov  edi,edi
```

Tamaño 2 bytes. Al ser menor que 6 continúa con la siguiente:

```
02  55       push   ebp
```

Tamaño 1 byte +2 bytes de la instrucción anterior = 3 bytes. Todavía sigue siendo menor que 6.

```
03  8BEC    mov  ebp,esp
```

Tamaño 2 bytes +3 bytes de las anteriores = 5 bytes. Continuamos:

```
04  64A118000000
  mov  eax,fs:[00000018]
```

Tamaño 6 bytes +5 bytes de las anteriores = 11 bytes. ¡Preparado!

Nuestro desensamblador de tamaño nos devolvería once. ¿Qué quiere decir esto? Pues que para un gancho de seis bytes, el número de bytes que deben ser copiados desde la *función destino* a la *Función Trampoline*, para que no se pierda ni se trunque ninguna instrucción ,es de once.

Análisis de los datos

Aquí tenemos el bloque principal del análisis. Hasta la línea 46 nos encontramos con un algoritmo que nos permitiría emular llamadas y saltos. Este algoritmo se basa en comprobar las distancias entre el EIP donde se ha producido la excepción con el anteior valor del registro. En caso de ser una distancia bastante considerable, nos encontramos ante una llamada o un salto, por lo que procederemos a restaurar el contexto para que apunte a la siguiente instrucción supervisada en vez de seguir a partir de la dirección donde nos llevó el salto o la llamada, también

Listado 6. *Crear proceso en modo suspendido y reservar memoria en despacio de direcciones*

```
Ret1 = CreateProcessA( szExecutableName, NULL, NULL, NULL,0,
  CREATE_SUSPENDED,NULL,NULL,
  &strStartupProceso,&strProceso);

 if( !Ret1 ) ShowError();
 printf("[OK]\n");


 printf("[+]Allocating remote memory...");
 Ret2 = (DWORD) VirtualAllocEx
( strProceso.hProcess,NULL,sizeof(HookCode),
  MEM_COMMIT,
  PAGE_EXECUTE_READWRITE);
```

sumaremos a nuestro contador los bytes que ocupa dicha instrucción.

En las lineas 47, 48, 49 se comprueba si tenemos las suficientes instrucciones analizadas como para alojar adecuadamente nuestro gancho.

Una parte fundamental del desensamblador son las líneas siguientes, 52, 53 y 54. En ellas activamos el *Trap Flag* poniendo a 1 el bit correspondiente en el registro *EFLAGS*. Esta es la base de una depuración paso a paso.

En menos de 256 bytes hemos construido un desensamblador de tamaño totalmente funcional. Creo que ya estamos listos para llevarlo a la práctica.

Aplicación práctica al análisis de Malware

Vamos a crear para nuestros propósitos un programa que inyectará y ejecutará código en el ejecutable que le pasemos como parámetro. Las técnicas de inyección de código en los procesos son bien conocidas. Existen varias formas pero todas se basan en prácticamente las mismas APIs.

- *VirtualAllocEx* para reservar un espacio de memoria en el proceso. En este espacio se inyectará el código. Para ello usaremos *WriteProcessMemory*.
- A la hora de ejecutar el código podemos elegir entre *CreateRemoteThread* o *SetThreadContext*.

Pero nosotros no vamos a usar ninguna de esas formas, sino que usaremos un método nuevo: *QueueUserAPC*.

Objetivos de la aplicación

Este pequeño programa inyecta en la calculadora de Windows un pequeño código que provoca la aparición de un *Message Box*. La calculadora tampoco aparecerá despues de ejecutarse este código. Imaginemos que en vez de inyectar un código inofensivo, inyecta un código malicioso perteneciente a un gusano. Imaginemos también que este pequeño programa ha sido empaquetado con un packer el cual incorpora varias protecciones antidebug y antidesensamblado. Necesitaríamos saber de manera rápida cómo logra inyectarse en otro ejecutable y qué código está inyectando. Para todo ello necesitaríamos urgentemente el desensamblado del ejecutable, pero como hemos dicho, incorpora un packer que nos está ralentizando el análisis debido a no poder usar el debugger fácilmente. Tampoco se queda residente en memoria el tiempo suficiente como para poderlo volcar con alguna herramienta de volcado de procesos (*ProcDump*...). De hecho, el ejecutable donde se inyecta, apenas se queda unas décimas de segundo en memoria, imposibilitándonos también el volcar su imagen. ¿Qué podemos hacer?

La solucion pasaría por hookear *ExitProcess* y de alguna manera mantenerle congelado en memoria al proceso (usando *Sleep*) el tiempo suficiente para poder volcarlo, y ya a posteriori, reconstruir el binario volcado con el fin de desensamblarlo y depurarlo con normalidad. ¿Por qué hookear *ExitProcess*? El 90% del malware empaquetado una vez ha alcanzado *ExitProcess* se encuentra totalmente desempaquetado en memoria. Nos podemos encontrar con excepciones en las cuales, sólo una parte de ejecutable está desempaquetado, pero esto no suele ser lo habitual ya que es algo complejo de diseñar. Nos centraremos en construir una herramienta que nos permita

Listado 7. *Reconstruimos ExitHook con la dirección obtenida mediante VirtuaAllocEx, parcheamos el Entry Point de la función destino,(en este caso ExitProcess) con ExitHook*

```
/*Reconstruimos ExitHook */
 *( DWORD* )( ExitHook + 1 ) = Ret2;

 printf("[OK]->Address : 0x%x",Ret2);
 printf("\n[+]Hooking %s...",argv[4]);
 printf("\n\t[-]Reading %d bytes from %s Entry Point ...", LenDasm, argv[4]);
 /* Copiamos las instrucciones supervisadas a la sección Trampoline */
 Ret1 =(DWORD) memcpy( (LPVOID)( HookCode + 27 ),(LPVOID)HookAddr, LenDasm);
 if( !Ret1 ) ShowError();
 printf("[OK]\n");
 printf( "\t[-]Hooking %s...", argv[4] );

 Ret1=0;
 while( !Ret1 )
 {
  ResumeThread(strProceso.hThread);
  Sleep(1);
  SuspendThread(strProceso.hThread);
  Ret1 = WriteProcessMemory(strProceso.hProcess, (LPVOID)HookAddr,  / *
Parcheamo la Función destino*/
  ExitHook, HookLength, NULL);
/* en memoria    */
 }

 printf("[OK]\n");

 printf("\t[-]Injecting Hook...");
 Ret1 = WriteProcessMemory(strProceso.hProcess,(LPVOID)Ret2,
/* Copiamos el código al espacio de direcciones*/
  HookCode, sizeof(HookCode), NULL);
/* del proceso recién creado    */

/* Dejamos correr al proceso */
ResumeThread(strProceso.hThread);
```

En la Red

- *http://www.reversemode.com/index.php?option=com_remository&Itemid=2&func=select&id=8*
  – Código fuente completo de todas las aplicaciones comentadas en el artículo. El desensamblador de tamaño. El ejemplo de malware y la aplicación para hookear.
- *http://research.microsoft.com/~galenh/dfPublications/HuntUsenixNt99.pdf*
  – Detours: Binary Interception of Win32 Functions.
- *http://msdn2.microsoft.com/en-us/library/ms253960(VS.80).aspx*
  – Structure Exception Handling in x86

Sobre el autor

Rúben Santamarta lleva desde los 16 años interesándose por el mundo de la ingeniería inversa, el bajo nivel y la seguridad informática en general. Con una formación totalmente autodidacta comenzó a trabajar a los 19 años como programador. Posteriormente, ha desarrollado su trabajo en sectores relacionados con el bajo nivel, los antivirus y las vulnerabilidades. Actualmente centra su actividad en este último campo.
Contacto con el autor: *ruben@reversemode.com*.

hookear rápidamente cualquier API de cualquier dll que esté usando el malware. Para ello obviamente, usaremos nuestro recién creado desensamblador de tamaño.

Como técnica de hooking usaremos Inline Hooking con una variación a medida de la técnica *Detour*.

HookCode contiene lo que sería el *Prólogo de la Funcion Detour*. Este prólogo consiste en avisarnos de que el malware ha alcanzado *ExitProcess* mediante una alarma auditiva, llamando a la API *Beep*. Posteriormente como habíamos comentado antes, llamaremos a Sleep con un parámetro pasado por la línea de comandos. Este parámetro será lo suficientemente alto como para permitirnos las operaciones de volcado o todas aquellas que quisieramos hacer. Una vez termina el prólogo, se ejecutarán las primeras instrucciones de *ExitProcess* (instrucciones supervisadas por el desensamblador de tamaño) y a continuación se devolverá el control a la siguiente instrucción que corresponda (*ExitProcess*+7).

Posteriormente deberemos reconstruir *HookCode* y *ExitHook* con las direcciones de memoria de las APIs y con los valores obtenidos por el desensamblador de tamaño.

Una pequeña reflexión para finalizar

Como hemos ido viendo a lo largo del artículo, en el extenso mundo de la ingeniería inversa, prácticamente todos sus campos acaban convergiendo en algun punto. Hemos combinado varias técnicas como desensambladores de tamaño, hooking e inyección de procesos para ayudarnos en el análisis de malware.

Paradójicamente, estas mismas técnicas son usadas por el malware para su propio beneficio, y es que la ingeniería inversa avanza para todos al mismo tiempo. Cuanto más complejidad alcanzan rootkits, virus, etc más concienzúdamente se estudian las técnicas usadas y cómo combatirlas.

Se crea de esta manera, una especie de carrera de obstáculos entre investigadores y programadores de malware o escritores de virus, aunque indudablemente millones de usuarios sufren las consecuencias, tampoco podemos negar que se fomenta la investigación y la innovación en ambos lados. ●



Análisis del tráfico en la Red

Bartosz Przybylski 

Grado de dificultad



Si administras cualquier red, puedes estar seguro de que tarde o temprano será objetivo del ataque. Sin embargo, eres capaz, si no de eliminar, por lo menos, de reducir la posibilidad de su éxito. Hay varios métodos de conseguirlo: desde la desactivación de los servicios, a través de cortafuegos, hasta los IDS. Puede, sin embargo, resultar que el problema más importante es la capacidad de diferenciar entre el tráfico de buenos y malos paquetes.

Pcap es una de las librerías para la codificación del tráfico de la red utilizadas con más frecuencia. Ofrece un acceso muy detallado a las capas respectivas ISO/OSI. Su ventaja es también su accesibilidad para varios sistemas operativos (más sobre el tema en hakin9 5/2005 no 14 en el artículo de Konrad Malewski *Pleno control, es decir, el acceso de bajo nivel a la red*) y en varios lenguajes de programación.

Martillo, destornillador, sniffer es decir, las herramientas de análisis

Al ocuparnos del análisis de la red, no es posible prescindir de algunas herramientas que pueden facilitar significativamente nuestro trabajo. Empecemos por los sniffers.

Ethereal

Ethereal es una de las herramientas más conocidas de análisis de la red. Dispone de muchas opciones que son muy útiles durante el análisis. Dos características dominantes de este proyecto son: la escritura del tráfico de red en varios formatos, y la interfaz gráfica. Aunque la segunda característica no es imprescindible para el análisis, constituye, sin embargo, un suplemento agradable que nos facilita el trabajo.

Tcpdump

Tcpdump es un sniffer muy bueno. Sus autores escribieron precisamente la librería *Pcap*. El programa posee algunos frontends no oficiales, sin embargo en principio fue diseñado, con la idea del uso directo a partir de la línea de comandos (shell) del sistema.

En este artículo aprenderás...

- Cómo analizar el tráfico en la red,
- Cómo a través del análisis descubrir los intentos del ataque,
- Cómo bloquear estos intentos.

Lo que deberías saber...

- Conocer las bases del funcionamiento de la red (ISO/OSI),
- Ser capaz de utilizar la shell de Linux.

Listado 1. *Verificación de la autenticidad de dos ficheros (aut.sh)*

```
#!/bin/sh
if [ -z $2 ]; then
echo "Usage: $0 authentic_file file_for_check";
exit
if

md5sum $1 | cut -c1-32 > /tmp/f1.cksum
md5sum $2 | cut -c1-32 > /tmp/f2.cksum
sha1sum $1 | cut -c1-40 >> /tmp/f1.cksum
sha1sum $2 | cut -c1-40 >> /tmp/f2.cksum
res=`/usr/bin/cmp /tmp/f1.cksum /tmp/f2.cksum'
if [ -z "$res" ]; then
echo "File is authentic"
else
echo "File is not authentic"
fi
rm /tmp/f1.cksum /tmp/f2.cksum
```

Listado 2. *Salida del programa capinfo*

```
$ capinfo traffic.cap
1   File name: traffic.cap
2   File type: libpcap (tcpdump, Ethereal, etc.)
3   Number of packets: 1194
4   File size: 93506 bytes
5   Data size: 213308 bytes
6   Capture duration: 342.141581 seconds
7   Start time: Thu Jun 23 14:55:18 2005
8   End time: Thu Jun 23 15:01:01 2005
9   Data rate: 623.45 bytes/s
10   Data rate: 4987.60 bits/s
11   Average packet size: 178.65 bytes
```

Nuestras armas

A continuación la lista de programas y scripts que vamos a utilizar para el análisis del tráfico de red:

- *capinfos* (parte del paquete ethereal),
- *tcpdstat*,
- *zonk.pl* (script sencillo para los administradores de red – escrito por el autor del artículo),
- algunos scripts propios.

Garantía de autenticidad

Si los resultados de análisis realizado deben servirnos como las pruebas contra el atacante, es importante comprobar la autenticidad del fichero con la transcripción del tráfico de red y del fichero en que fue basado nuestro análisis.

Es importante que el fichero original que servirá como la prueba principal tenga la fecha de creación posiblemente más cercana a la fecha de la intrusión. Para asegurarlo, hay que copiar el fichero con el tráfico hasta un directorio separado y atribuir al fichero y al directorio los derechos solamente de lectura (*read-only*). Lo hacemos del modo siguiente:

```
mkdir ~/analyze
cp ./traffic.cap ~/analyze
chmod 444 ~/analyze/
traffic.cap ~/analyze/
```

Cuando tenemos asegurada la copia en la que serán basadas las pruebas, debemos comprobar también su autenticidad, en lo cual nos ayudará el script sencillo del Listado 1.

Este script puede ser útil durante la verificación de la autenticidad del tráfico interceptado. Vale también la pena guardar los totales de control para el fichero original, lo cual será más convincente.

Análisis de red

Pasemos ahora a nuestra tarea principal, es decir, el análisis del tráfico interceptado. Para poder realizar este análisis hay que recoger la información básica sobre el tráfico interceptado. Con este fin vamos a utilizar el programa *capinfo*, que es un elemento del paquete *ethereal*.

Analicemos el Listado 2 y pensemos qué tipo de información obtendremos gracias a *capinfo*.

La primera línea (*File name*) la omitimos, ya que contiene el nombre del fichero. En la segunda tenemos la información sobre el formato del fichero. En este caso es el fichero en el formato pcap. Otro sistema de escritura del tráfico también popular es *Microsoft Network Monitor* x.x donde x.x es la versión de la librería. La versión más frecuente es la versión 2.x (por supuesto, no son los formatos únicos de escritura de los ficheros de tráfico en la red).

La línea siguiente de salida (3) contiene el número de paquetes que *atravesaron* nuestra red durante el sniffing. Luego tenemos (4) el tamaño del fichero y (5) tamaño de datos del tráfico guardado. Hay también: (6) la duración precisa de la intercepción (7), fecha de inicio y (8) fin de sniffing, (9) el flujo medio de datos en bytes y (10) en bits, mientras que la última línea presenta (11) el tamaño medio del paquete.

Ya en el momento de análisis podemos hacer algunas conclusiones respecto al tráfico inautorizado eventual (se trata sobre todo de grandes empresas). Si en un intervalo del tiempo pequeño (el valor no muy elevado de *Capture duration*) aparece la cantidad sospechosamente grande de paquetes o el tamaño medio de paquetes es grande, podemos sospechar que en nuestra red son utilizados los programas que ayudan la descarga del Internet. Sin embargo, esto no tiene porque ser siempre verdad, este tráfico puede ser también debido, por ejemplo, a la descarga de las actualizaciones del software.

El paso siguiente es el reconocimiento más detallado del tráfico, esta

Pedido de suscripción






Por favor, rellena este cupón y mándalo por fax: 0048 22 887 10 11 o por correo: Software-Wydawnictwo Sp. z o. o., Piaskowa 3, 01-067 Varsovia, Polonia; e-mail: *suscripcion@software.com.pl*

Nombre(s) .... Apellido(s) ....

Dirección ....

C.P. .... Población ....

Teléfono .... Fax ....

Suscripción a partir del Nº ....

e-mail (para poder recibir la factura) ....

☐ Renovación automática de la suscripción

Título	número de ejemplares al año	número de suscripciones	a partir del número	Precio
Sofware Developer's Journal Extra! (1 CD-ROM) – el antiguo Software 2.0 Bimestral para programadores profesionales	6			38 €
Linux+DVD (2 DVDs) Mensual con dos DVDs dedicado a Linux	12			86 €
Hakin9 – ¿cómo defenderse? (1 CD-ROM) Bimestral para las personas que se interesan de la seguridad de sistemas informáticos	6			38 €
Linux+ExtraPack (7 CD-ROMs) Las distribuciones de Linux más populares	6			50 €
			En total	

Realizo el pago con:

☐ tarjeta de crédito (EuroCard/MasterCard/Visa/American Express) nº ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ CVC Code ☐☐☐

Válida hasta ☐☐☐☐

☐ transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO
Número de la cuenta bancaria: 0049-1555-11-221-0160876
IBAN: ES33 0049 1555 1122 1016 0876
código SWIFT del banco (BIC): BSCHESMM

Fecha y firma obligatorias:

vez al nivel de paquetes. A nosotros nos interesa el tipo de protocolo y el tamaño del paquete. Conoceremos también la característica más detallada de la carga de la conexión. Con este fin vamos a utilizar el programa modificado de Dave Dittrich *tcpdstat*. Vamos a basarnos en el ejemplo del Listado 3.

Podemos leer de él los datos básicos parecidos a los que obtuvimos ya antes con el uso del programa `capinfos`. Está vez, sin embargo, disponemos también de más información. Se trata del número de paquetes en un intervalo dado de tamaño, y también de las informaciones más detalladas acerca de los protocolos del tráfico (esta información está en la parte *Protocol Breakdown*). Las líneas respectivas fueron guardadas en el formato siguiente:

```
["nivel" del protocolo] protocolo
número_paquetes (porcentaje_de_todos_
los_paquetes) número_bytes (porcentaje_
del_total) número_medio_bytes_por_
paquete
```

Gracias a esta información podemos comprobar qué paquetes de las familias respectivas de protocolos atraviesan nuestra red. Debemos fijarnos en esto ya que en base a los protocolos existentes podemos deducir si nuestra red es objetivo del análisis remoto o del ataque realizado en un momento determinado.

Fijémonos en la afluencia significativa de paquetes TCP. Si la información sobre los paquetes HTTP no deben inquietarnos (a no ser que sea el uso inautorizado de nuestras aplicaciones PHP), debemos pensar qué es lo que provoca el tráfico tan grande de otros paquetes TCP (other). Es un tráfico definido en pcap a base de la capa de red, pero no definido a base del puerto. Con más frecuencia es el tráfico output, sin embargo, no necesariamente. Puede ser también el intento de escanear nuestro sistema.

Otra información devuelta por tcpdstat son, entre otros, 10 cargas más grandes de la red. Sin embargo, esta información es necesaria sola-

Listado 3. *Salida del programa tcpdstat*

```
DumpFile: traffic.cap
FileSize: 0.09MB
Id: 200506231455
StartTime: Thu Jun 23 14:55:18 2005
EndTime: Thu Jun 23 15:01:01 2005
TotalTime: 342.14 seconds
TotalCapSize: 0.07MB CapLen: 68 bytes
# of packets: 1194 (208.31KB)
AvgRate: 5.08Kbps stddev:30.22K
### IP flow (unique src/dst pair) Information ###
# of flows: 66 (avg. 18.09 pkts/flow)
Top 10 big flow size (bytes/total in %):
20.0% 16.3% 15.7% 12.9% 4.8% 4.0% 2.9% 1.3% 1.3% 1.2%
### IP address Information ###
# of IPv4 addresses: 68
Top 10 bandwidth usage (bytes/total in %):
69.9% 21.5% 18.5% 17.5% 16.9% 13.9% 5.4% 5.2% 4.5% 4.3%
# of IPv6 addresses: 4
Top 10 bandwidth usage (bytes/total in %):
81.5% 59.2% 40.8% 18.5%
### Packet Size Distribution (including MAC headers) ###
<<<<
[ 32- 63]: 857
[ 64- 127]: 104
[ 128- 255]: 79
[ 256- 511]: 61
[ 512- 1023]: 14
[ 1024- 2047]: 79
### Protocol Breakdown ###
<<<<
protocol packets bytes bytes/pkt
------------------------------------------------
[0] total 1194 (100.00%) 213308 (100.00%) 178.65
[1]  ip 988 ( 82.75%) 198381 ( 93.00%) 200.79
[2]   tcp 884 ( 74.04%) 180408 ( 84.58%) 204.08
[3]    http(s) 219 ( 18.34%) 124825 ( 58.52%) 569.98
[3]    other 665 ( 55.70%) 55583 ( 26.06%) 83.58
[2]   udp 94 ( 7.87%) 17247 ( 8.09%) 183.48
[3]    dns 9 ( 0.75%) 2752 ( 1.29%) 305.78
[3]    other 85 ( 7.12%) 14495 ( 6.80%) 170.53
[2]  icmp 7 ( 0.59%) 546 ( 0.26%) 78.00
[2]  igmp 3 ( 0.25%) 180 ( 0.08%) 60.00
[1] ip6 5 ( 0.42%) 422 ( 0.20%) 84.40
[2]  icmp6 5 ( 0.42%) 422 ( 0.20%) 84.40
```

```
▽ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    Host: google.pl\r\n
    User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.12) Gecko/20051221 Firefox/1.0.7\r\n
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/pr
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie: PREF=ID=9f6aa6ad138e14e6:TM=1135184128:LM=1135184128:S=Xo58VqtTTwE-e4ql\r\n
    \r\n

0060  6c 0d 0a 55 73 65 72 2d  41 67 65 6e 74 3a 20 4d   l..User- Agent: M
0070  6f 7a 69 6c 6c 61 2f 35  2e 30 20 28 58 31 31 3b   ozilla/5 .0 (X11;
0080  20 55 3b 20 4c 69 6e 75  78 20 69 36 38 36 3b 20    U; Linu x i686;
0090  65 6e 2d 55 53 3b 20 72  76 3a 31 2e 37 2e 31 32   en-US; r v:1.7.12
00a0  29 20 47 65 63 6b 6f 2f  32 30 30 35 31 32 32 31   ) Gecko/ 20051221
00b0  20 46 69 72 65 66 6f 78  2f 31 2e 30 2e 37 0d 0a    Firefox /1.0.7..
00c0  41 63 63 65 70 74 3a 20  74 65 78 74 2f 78 6d 6c   Accept:  text/xml
00d0  2c 61 70 70 6c 69 63 61  74 69 6f 6e 2f 78 6d 6c   ,applica tion/xml
```

Figura 1. *Descarga de documentos de la web – ejemplo*

mente en el caso de la creación de los datos estadísticos.

Ahora, cuando tenemos ya la información sobre los paquetes y protocolos, hay que saber más sobre los destinatarios y remitentes de los paquetes. Para este fin nos servirá un script sencillo del Listado 4.

Este script en su salida nos mostrará las direcciones IP de los paquetes que pasan por nuestra estación (hay que tener en cuenta que son solamente las direcciones de origen). Ya a base de estos datos podemos determinar las reglas de cortafuegos para el rechazo o aceptación de los datos IP. Hay que verificar también si alguna dirección no pertenece a *bogone space*.

Si es así, esto significa que hubo o hay un intento de ataque DDos en nuestra red. En este caso haría falta utilizar (durante algunos días) el script que bloquearía en el cortafuegos las direcciones bogone (un script así está disponible en la página *http://completewhois.com/*).

Una pequeña modificación en `searchip` nos facilitará la información sobre las direcciones MAC de las interfaces de las que provienen los paquetes analizados. Sin embargo, esto no nos hace falta aquí ya que analizamos el tráfico entrante de Internet, y no aquél que proviene de la red local.

En cuanto a los ataques DoS y DDoS, es un tema tan amplio que sería necesario un artículo aparte sobre el tema. En hakin9 no 5/2004 Andrzej Nowak y Tomasz Potęga describen cómo protegerse contra los ataques.

Hay que añadir que el bloqueo de los ataques DDoS, que tuvieron lugar ya no tiene mayor sentido ya que los atacantes seguramente utilizarán otro pool de direcciones.

Listado 4. *Script para la búsqueda de varios tipos de ip del fichero pcap (searchip.pl)*

```
#!/usr/bin/perl
use Switch;
use Net::Pcap;
my $err;
my $rep;
my $curr_ip;
my @ip_table = ();
sub connread;
if ($ARGV[0] eq "")
{
print "Usage: ./search_ip
<filename/filepath>";
exit;
}
if($pcap = Net::Pcap::open_offline($ARGV[0], \$err))
 # abrimos el fichero
{
Net::Pcap::loop($pcap, -1, \&connread, '');
 # transmite cada paquete a la función connread
}
else
{
print "Error\n";
exit;
}
print "Founded different IP:\n";
foreach $ip_table(@ip_table)
{
print $ip_table."\n";
}
sub connread # la función principal
{
my($data, $header, $packet) = @_;
my $packet = unpack('H*', $packet);
$rep=0;
# busca en el paquete ip
if ($packet =~ m/^\w{44}(..)(..)(\w{4}|
\w{8})(..)(..)(..)(..)\w{8}(....)(....)\w+(.*)/)
{
# guarda en la variable y compara con los ya añadidos
$curr_ip = hex($4).".".hex($5).".".hex($6).".".hex($7);
foreach $ip_table(@ip_table)
{
if($ip_table eq $curr_ip)
{
$rep = 1;
}
}
if ($rep == 0) {
# si un ip así no existe todavía, añádelo a la tabla
push(@ip_table, $curr_ip);
}
}
}
```

HTTP, FTP – análisis de información

Preguntémonos ahora si somos capaces de prevenir el análisis detallado de nuestro servidor web y los efectos posteriores de una operación así. ¿Es posible prever el uso de nuestro FTP para el escaneo de los puertos? La respuesta es *sí*, sin embargo, no es fácil. Haciéndolo *a mano* podemos tardar mucho tiempo. Hagamos un fragmento de este análisis para ver que consume bastante tiempo.

Abramos nuestro fichero con el tráfico registrado con ayuda del paquete *ethereal* (lo utilizamos sobre todo teniendo en cuenta la interfaz gráfica). Escogemos uno de los paquetes que empiezan la descarga de documentos desde las páginas web

Listado 5a. *fhelp.pl*

```
#!/usr/bin/perl
use Switch;
use Net::Pcap;
my $dev = "eth0";
 # interfaz para escuchar
my $pcap;
my $err;
my $packet;
sub connread;
if ($ARGV[0] eq "-h") {
print "Usage: ./fhelp.pl <filename>\n\r
If no filename given will start live capture\n";
exit;
}
# comprobamos si al fichero le fue
# asignado un nombre, si es así, lo abrimos,
# en el caso contrario abrimos "live capture"
if ($ARGV[0] == "") {
if($pcap = Net::Pcap::open_live
($dev, 2000, 1, 1000, \$err)) {
Net::Pcap::loop($pcap, -1, \&connread, '');
}
else {
print "Error\n$err\n";
exit;
}}
else {
if($pcap = Net::Pcap::
open_offline($ARGV[0], \$err)) {
Net::Pcap::loop($pcap, -1, \&connread, '');
}
else {
print "Error\n$err\n";
exit;
}}
sub connread {
my ($data, $header, $packet) =
@_; $get = "n";
my $packet = unpack('H*', $packet);
# descargamos y descomprimimos el paquete
# buscamos las conexiones con los puertos 80 y 21
if ($packet =~ m/^\w{44}(..)(06)(\w{4}|
\w{8})(..)(..)(..)(..)\w{8}(....)(0050|0015)(.*)/) {
$curr_ip = sprintf("%d.%d.%d.%d",
hex($4), hex($5), hex($6), hex($7));
$traffic = sprintf("%s",$10);
# buscamos los paquetes http incompletos
if ($traffic =~ /(474554|48454144)/) {
```

(el paquete debe contener la frase GET o HEAD).

El ejemplo de un paquete así lo tenemos en la Figura 1. Como podemos observar cada *buena* conexión dispone de una información enviada automáticamente por el navegador. Se trata, entre otros, del tipo y de la versión del navegador (User-agent), el tipo de ficheros aceptado (Accept), codificación (Accept-Encoding), lenguaje preferido (Accept-Language). Además, puede aparecer la información si la conexión debe ser mantenida, durante cuánto tiempo e ID cookies.

Prácticamente todos los atacantes de red durante el análisis de la información puesta a la disposición por el servidor (sus banners) no facilitan los datos adjuntos por defecto por el navegador (vista la eliminación por el servidor de las conexiones inactivas durante mucho tiempo). Este hecho puede ser de mucha ayuda.

Si analizando el tráfico encontramos las cabeceras incompletas, podemos suponer que nuestro servidor fue ya objeto de un análisis y podemos esperar también un ataque. Desgraciadamente, o por suerte, no siempre es verdad, ya que los navegadores de consola (por ejemplo, *Lynx, links*) no dejan su "tarjeta de visita". Así podemos equivocarnos. Sin embargo, dejan la información sobre los cookies, lo que distingue estos navegadores de los agresores que estudian los banners del servidor.

Ocupémonos ahora de FTP. Si ponemos a nuestra disposición un servidor FTP anónimo, éste puede ser empleado para el escaneo de los puertos en cualquier servidor. Es posible con ayuda del comando PORT. Por supuesto podemos desactivarlo, pero esto provocará también la imposibilidad de realización de las conexiones activas con nuestro servidor. Sin embargo, si por cualquier razón necesitamos las conexiones activas, haría falta realizar de vez en cuando el análisis del tráfico FTP en nuestro servidor. Como en el caso de http podríamos analizar el tráfico y bloquear en el cortafuegos las direcciones IP respectivas, sin embargo, nos interesa su eficacia. De ayuda nos será, entonces, el script *fhelp.pl* (véase el Listado 5).

Este sencillo script analiza un fichero determinado en el formato pcap, o trabajo *en vivo*, encontrando en los paquetes http los casos de

Bogon space

En resumen, *bogon space* es el espacio de direcciones en Internet que no fueron todavía asignados a los propietarios. Y los paquetes de unas direcciones así deben ser considerados inadecuados y eliminados.

La lista bogon puesta al día regularmente se encuentra en la página *http://completewhois.com.*

Listado 5b. *fhelp.pl*

```
if (!($traffic =~ m/^(.*)\w
(20485454502f312e(31|30)0d0a)(.*)
(557365722d4167656e743a)(.*)/)) {
print "incomplete
http header from $curr_ip\n";
print "do you want to
block ip $curr_ip on iptables? [y/N]: ";
$get = <>;
if (($get eq "y")|($get eq "Y")) {
system("iptables -A INPUT -
p tcp -s $curr_ip -j DROP");
}
}}
# buscamos el comando
# port en los paquetes ftp
if ($traffic =~ /(706f7274|504f5254)/) {
print "PORT command used in
ftp connection from $curr_ip\n";
print "do you want to block ip
$curr_ip on iptables? [y/N]: ";
$get = <>;
if (($get eq "y")|($get eq "Y")) {
system("iptables -A INPUT -
p tcp -s $curr_ip -j DROP");
}}
}}
```

falta de información sobre el navegador del cliente. Analiza también el tráfico FTP en búsqueda del comando PORT. Si encuentra un tráfico así, pregunta al usuario si bloquear la dirección correspondiente IP en el cortafuegos (colabora solamente con iptables). Es evidente, que debemos activar el script como el usuario root.

SSL – el talón de Aquiles

Todos los que realizan los análisis de red, tarde o temprano darán con el tráfico cifrado. Es el problema clásico de la mayoría de análisis.

El desciframiento de este tráfico es posible pero según muestra la práctica, es completamente improductivo. En primer lugar, en 999 de 1000 casos es el tráfico autorizado. En segundo lugar, el desciframiento de los paquetes tomaría muchísimo tiempo, ya que sería necesario encontrar la llave de secuencia para cada transmisión. Si pensamos detalladamente en este tipo de transmisión, pronto llegaremos a la conclusión que la única solución razonable es la autorización de la conexión cifrada solamente para los *hosts* de confianza. Desgraciadamente, esto no funciona, si nuestro servidor debe estar disponible para el público y ofrecer, en principio, las conexiones cifradas a los clientes de sus servicios.

Trabajadores perezosos

En este momento vale la pena recordar el script perl *zonk.pl* (del autor del artículo).

Esta aplicación sencilla basada en las expresiones condicionales y la librería Pcap busca en el tráfico corriente de red los abusos sencillos de la conexión (tráfico p2p, im, irc) a base de los números de puertos de los servidores utilizados por *las transmisiones prohibidas*. El script puede ser útil para los administradores de las redes corporativas, ayudando a detectar y eliminar el uso excesivo e inautorizado de los recursos de la empresa por los mismos trabajadores. Zonk, igual que otros scritps mencionados en el artículo, están en la página del autor (véase el marco En la red). Allí podéis encontrar también otras herramientas útiles de administrador.

¿Para qué todo esto?

Teniendo en cuenta muchas restricciones mencionadas en el artículo, podemos plantearnos la pregunta, si vale la pena analizar el tráfico en nuestra red.

Seguramente esto no será remedio para todos nuestros problemas, sin embargo, no podemos subestimar la importancia de la información resultante de un buen análisis realizado en el tiempo adecuado. De este modo podemos también detectar los ataques *brute force*, DoS y DDoS, el tráfico p2p e incluso el uso excesivo de las conexiones http.

Por supuesto, lo más difícil es determinar el *tiempo adecuado*, en el que hay que realizar el análisis. Por eso, vale la pena realizar los análisis regulares y utilizar los scripts que son capaces detectar automáticamente mayores anomalías. Tengamos también en cuenta que el elemento más importante del análisis realizado no es recoger los datos sino la interpretación adecuada del tráfico interceptado.

Hay que tener en cuenta también que después de utilizar todos los scripts y herramientas nos hará falta repasar todo el tráfico con el fin de detectar las anomalías más detalladas, no obstante, es una de las consecuencias que deben tener en cuenta todos los analistas de la red. ●

En la Red

- *http://www.ethereal.com/* – página del proyecto sniffer ethereal,
- *http://www.tcpdump.org/* – página libpcap y tcpdump,
- *http://www.netfilter.org/* – página del proyecto iptables.
- *ftp://tracer.csl.sony.co.jp/pub/mawi/tools/* – aquí podéis descargar tcpdstat
- *http://aqu.banda.pl/scripts/network_analyzing/* – aquí podéis descargar zonk.pl



Problemas con autenticación HTTP

Emilio Casbas 

Grado de dificultad



El protocolo HTTP, nos ofrece un mecanismo de autenticación desafío-respuesta que puede ser usado por un servidor web o servidor proxy para permitir o denegar el acceso a recursos web.

Hoy día, sobre la web se realizan millones de transacciones y accesos a datos privados y confidenciales. La web facilita todo esto, pero tambien es necesario seguridad, necesitamos saber quien accede a nuestros datos sensibles, y quien puede realizar operaciones con privilegios.

Necesitamos saber, que los usuarios no autorizados, no pueden ver documentos para los cuales no tienen acceso,

Los servidores necesitan saber de alguna manera quien es cada usuario, y en base a ello decidir que tipo de acciones pueden realizar.

La autentificacion es una técnica de identificación basada en el conocimiento, es decir, algo que el usuario conoce, tal como una password o un PIN.

HTTP provee una funcionalidad nativa para la autenticación HTTP

Realmente HTTP define dos protocolos de autentficación oficiales: *autenticación básica* y *autenticación digest*. Aquí, me centraré especialmente en el método de *autenticación básica*, que es el más ampliamente utilizado por clientes y servidores web, y el menos seguro.

Ambitos de aplicación de este método de autenticación

Servidores web en internet. Aquí estaríamos en la situación más corriente. Un usuario desde su casa con su conexión normal o desde un cibercafé, accede a un servidor web que tiene configurado la autenticación HTTP para acceder a ciertas zonas de él. Haciendo un pequeño repaso por algunas web corporativas, podemos ver gran cantidad de sitios que utili-

En este artículo aprenderás...

- Diferentes ámbitos de la autenticación HTTP
- Diferencias de la validación HTTP en los diferentes ámbitos.
- Ejemplos prácticos de conversaciónes HTTP
- Debilidad de la autenticación.
- Soluciones o alternativas.

Lo que deberías saber...

- Modelo OSI
- Conocimiento del protocolo HTTP.

zan este tipo de autenticación para acceder a partes de acceso restringido en la web.

Servidores web en intranet. Aquí el ámbito de aplicación es menor, ya que esta restringido sólo a la intranet de la empresa, pero los problemas asociados con este tipo de autenticación son los que se han comentado anteriormente, cualquier recurso disponible dentro de la red será susceptible.

Servidores proxy en internet. Tambien se puede dar el caso que por ejemplo para navegar por algunos recursos de una web determinada, sólo se pueda hacer a través de un servidor proxy de esa institución o para controlar cualquier tipo de acceso, por ello la autenticación HTTP tambien puede estar implementada en el proxy y todos los datos pasarían tambien por internet.

Servidores proxy en intranet. Tambien es muy común que dentro de redes corporativas, la única forma de poder acceder a Internet es a través de un servidor proxy con la finalidad de controlar todo el uso que se hace de Internet. Para ello la configuración de un servidor proxy con validación para controlar que usuarios acceden es lo normal en estos casos. Normalmente este tipo de validación estará integrada con lo que ya existe dentro de esa intranet. Agravando el problema del Single Sign On que ya veremos más adelante.

Breve Introducción a la autenticación digest

La finalidad de la autenticación digest, es no enviar nunca el password a través de la *red* para ello envía al servidor un *resumen* o *huella* de el password de una manera irreversible.

La autenticación digest, fue desarrollada de forma compatible y como una alternativa más segura a la autenticación básica, pero no es uno de los protocolos denominados seguros comparados con aquellos que utilizan mecanismos de clave-pública (SSL) o mecanismos de intercambio de tickets (kerberos). La autenticación digest no posee una fuerte autenticación ni ofrece protección de confidencialidad fuera de la protección del password, el resto de la petición y respuesta van en texto plano.

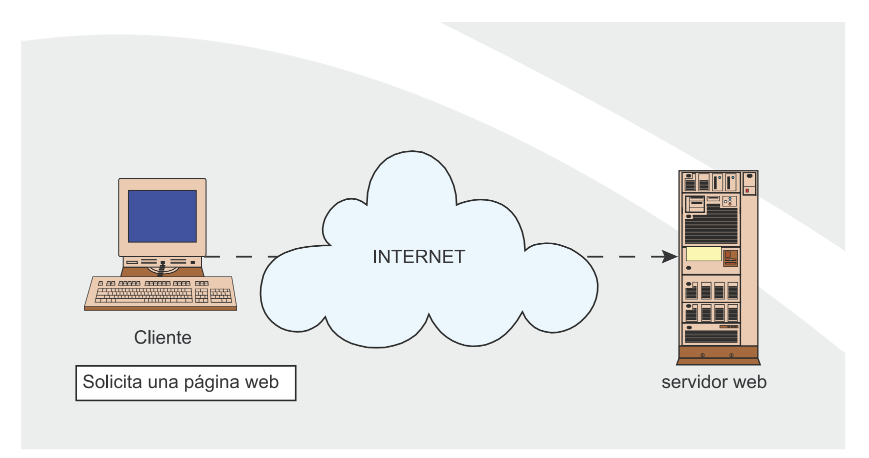


Figura 1. *Servidores web en Internet*

Autenticación básica

La autenticación básica es uno de los protocolos de autenticación HTTP más utilizados. La mayoría de los clientes y servidores web la implementan. Lo mejor antes que nada es ver una descripción gráfica de lo que es.

A continuación explicamos más detalladamente los pasos anteriores de la autenticación HTTP:

- Un usuario solicita un recurso web (por ejemplo una imagen)
- El servidor comprueba que es un recurso protegido y le envía al cliente un desafío de password con la cabecera HTTP *Authorization Required* y código 401.
- El navegador del usuario recibe el código y la cabecera de autorización y muestra el diálogo de usuario/contraseña. Cuando el usuario introduce los datos, el navegador realiza una codificación en base64 con los datos introducidos y lo reenvía al servidor en la cabecera del cliente *Authorization*.
- El servidor decodifica el nombre de usuario y password, y comprueba que tiene acceso al recurso protegido.

Como se puede comprobar, la autenticación básica transmite el par *usuario:password* de forma no encriptada desde el navegador al servidor y como tal, no debería ser usado para logins sensibles a menos que se este operando sobre un medio encriptado tal como SSL.

Ejemplo práctico

A partir de aquí, ya sabemos que es la autenticación HTTP, y que pasos sigue, ahora veamos esos pasos

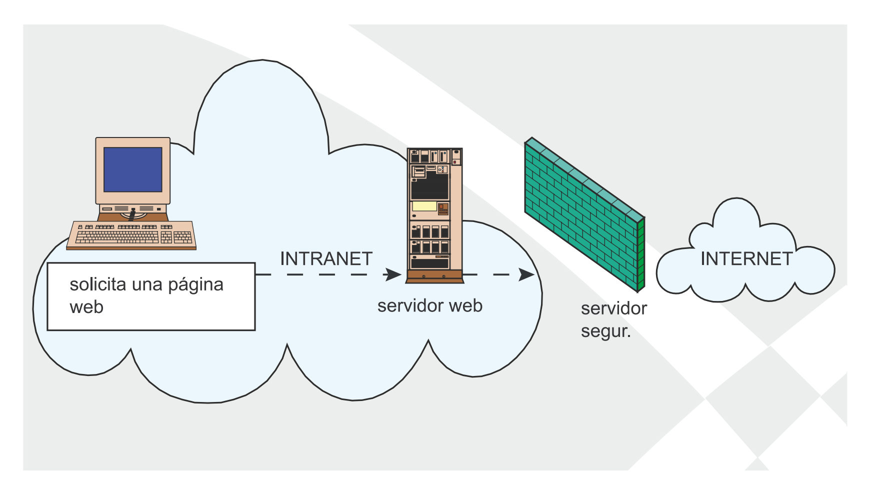


Figura 2. *Servidores web en Intranet*

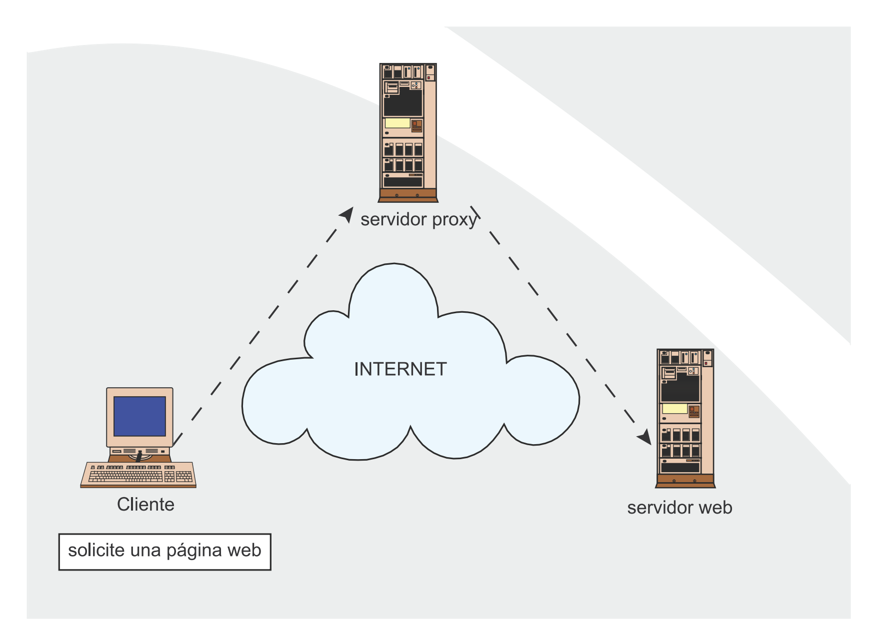


Figura 3. *Servidores Proxy en Internet*

más detenidamente y de una forma práctica con nuestro navegador web mozilla. Para ello, necesitaremos una extensión para capturar las cabeceras HTTP que realizamos.

La extensión que necesitamos se llama *livehttpheaders* y podemos descargarla desde *http://livehttpheaders.mozdev.org/*. La instalamos siguiendo los pasos que pone en la página, y a continuación nos aparecerá en mozilla la opción – ver Figura 6. Pulsamos sobre esa nueva opción, y nos aparecerá el recuadro – ver Figura 7.

A partir de ahora, obtendremos toda la información de las cabeceras HTTP de todos los sitios por los que navegemos, de esta forma, hemos conseguido capturas de las cabeceras que a continuación se explican.

Cabeceras y Explicación

El cliente envía una petición HTTP estándar solicitando un recurso:

```
    GET
/doc/ecasbas/ HTTP/1.1\rn
    Host: www.prueba.es\rn
    User-Agent: Mozilla/5.0
(Windows; U;
Windows NT 5.1; en-US; rv:1.7.12)
    Accept: text/xml,text/html;q=
0.9,text/plain;q=0.8,
image/png,*/*;q=0.1\rn
    Accept-Language:
 en-us,en;q=0.7,es;q=0.3\rn
    Accept-Encoding: gzip,deflate\rn
    Accept-Charset: ISO-8859-1,
utf-8;q=0.7,*;q=0.7\rn
    Keep-Alive: 300\rn
    Connection: keep-alive\rn
```

El servidor lee su archivo de configuración y determina que el recurso solicitado esta protegido con contraseña. El servidor sólo puede permitir el acceso a usuarios conocidos.

El servidor entonces le contesta al cliente con una respuesta requerida de autorización indicándole al cliente con el código HTTP 401:

```
    HTTP/1.0 401 Unauthorized\rn
    Date:
 Mon, 16 Jan 2006 11:17:51 GMT\rn
    Server: Apache/2.0.55
(Unix) mod_ssl/
2.0.55 OpenSSL/0.9.7g PHP/5.1.1\rn
    WWW-Authenticate:
 Basic realm=
"ByPassword"\rn
    Accept-Ranges: bytes\rn
    Content-Length: 3174\rn
    Content-Type: text/html\rn
    X-Cache: MISS from
www.prueba.es\rn
    Connection: keep-alive\rn
```

El navegador del cliente, interpreta este código HTTP 401 como un desafío de autenticación, y el navegador entonces muestra el prompt de *usuario:password* mostrando el nombre del host y el realm – ver Figura 8.

El cliente reenvía la petición con el *usuario/password* introducidos en el prompt anterior:

```
    GET /doc/ecasbas/ HTTP/1.1\rn
    Host: www.unav.es\rn
    User-Agent: Mozilla/5.0
(Windows; U;
 Windows NT 5.1; en-US; rv:1.7.12)
    Accept: text/tml+xml,text/
html,image/jpeg,
image/gif;q=0.2,*/*;q=0.1\rn
    Accept-Language:
 en-us,en;q=0.7,es;q=0.3\rn
    Accept-Encoding: gzip,deflate\rn
    Accept-Charset: ISO-8859-1,
utf-8;q=0.7,*;q=0.7\rn
    Keep-Alive: 300\rn
    Connection: keep-alive\rn
```

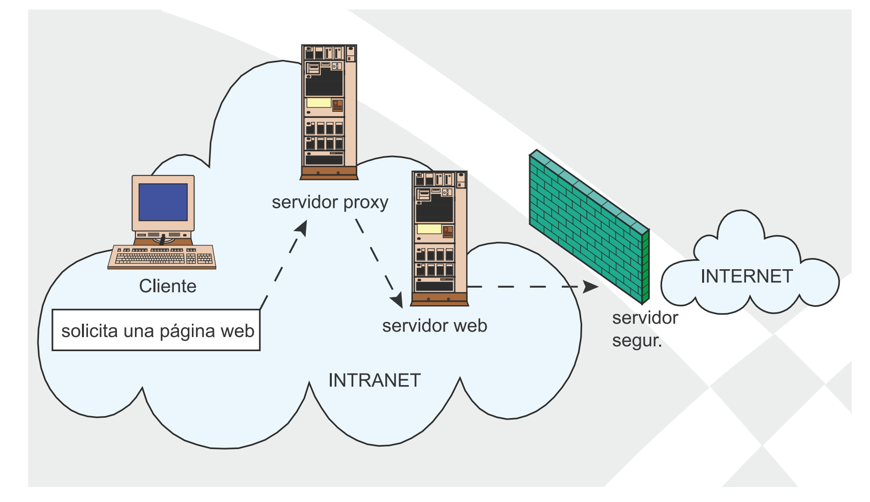


Figura 4. *Servidores Proxy en Intranet*

```
    Authorization: Basic
ZWNhc2JhczpwcnVlYmE=\rn
```

El servidor compara la información del cliente con su lista de usuarios/ passwords. Si la autorización falla, el servidor volverá a mandarle la cabecera de autenticación requerida HTTP 401. Si los datos introducios son correctos, el servidor mostrará el recurso solicitado.

El servidor da paso al recurso solicitado:

```
    HTTP/1.0 200 OK\rn
    Date:
 Mon, 16 Jan 2006 11:17:58 GMT\rn
    Server: Apache/2.0.55
 (Unix) mod_
ssl/2.0.55 OpenSSL/0.9.7g PHP/5.1.1\rn
    Last-Modified:
 Fri, 13 Jan 2006 10:31:02 GMT\rn
    ETag: "125b019-5-f636a580"\rn
```

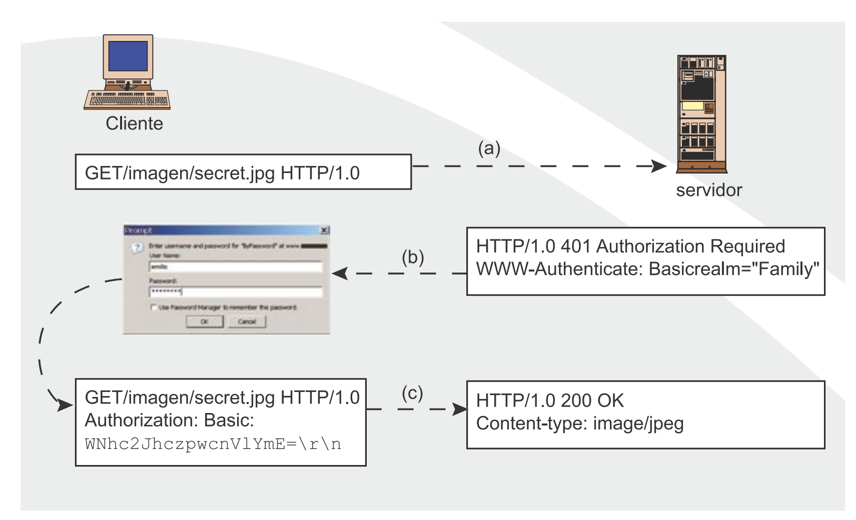


Figura 5. *Autenticación básica*

```
    Accept-Ranges: bytes\rn
    Content-Length: 5\rn
    Content-Type: text/html\rn
    X-Cache: MISS from
www.prueba.es\rn
    Connection: keep-alive\rn
```

En los campos anteriores, hemos visto campos especiales que han sido añadidos a varias cabeceras HTTP. En el paso 3, cuando el servidor envía la respuesta con la cabecera 401, incluye un campo especial.



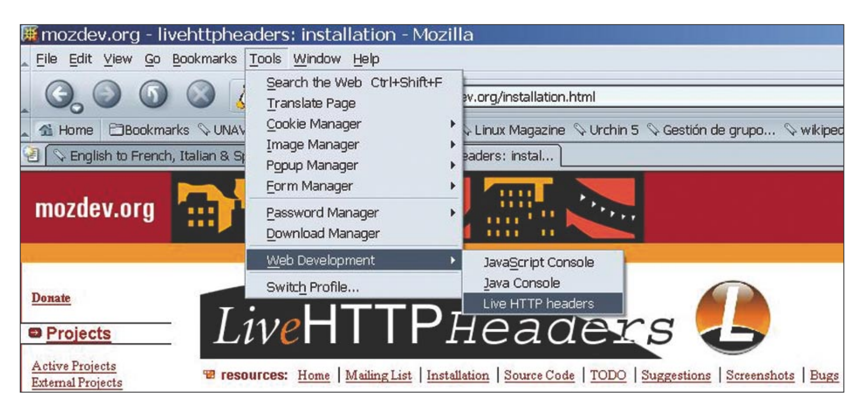


Figura 6. *Extensión livehttpheaders*

```
WWW-Authenticate:
 Basic realm="ByPassword"\rn
```

El valor *Basic* muestra que estamos pidiendo al browser usar autenticación básica. La información de la cadena *realm* es una cadena arbritaria enviada para mostrar al usuario un recordatorio del tipo de autenticación. La imagen del punto 4 muestra como la caja de diálogo de mozilla pide la autenticación mostrando el realm y el host.

El usuario rellena el formulario y lo envía. El navegador automáticamente reenvía la petición como vemos en el paso 5. Aquí es donde vemos que se han añadido algunos campos en la petición HTTP estándar.

```
Authorization:
 Basic ZWNhc2JhczpwcnVlYmE=\rn
```

Aquí es donde el navegador web envía la información de la autorización actual al servidor. El campo *Authorization* se ve que está compuesto por dos valores. La palabra *Basic* muestra como el login esta siendo enviado de acuerdo al método de autenticación básica. El bloque de datos que le sigue es el actual login enviado por el navegador. Nuestros datos de login no aparecen directamente, pero no es una rutina de encriptación, es una codificación en base 64.

A modo resumen, la codificación en *base64* representa secuencias arbitrarias de octetos de una forma no necesariamente legible por los humanos. Los algoritmos de codificación y decodificación son simples pero los datos codificados suelen ser un 33% más grandes que los datos sin codificar.

Para más información sobre esta codificación consultar en los enlaces del final del documento.

Código perl para decodificar una cadena en base64 como la anterior.

```
--codigo perl--
#!/usr/bin/perl
use MIME::Base64;
while (<>) {
  print MIME::
Base64::decode_base64($_);
}

--
```

Con el código anterior, el login en texto plano, puede ser trivialmente decodificado al subyacente formato de *usuario:password*.

```
ZWNhc2JhczpwcnVlYmE=
 --> base64Decode() -->
 "ecasbas:prueba"
```

La implementación de la autenticación *digest* es exáctamente el mismo proceso que la autenticación básica anterior, la única diferencia está en el número de argumentos suministrados por el navegador y en el formato de el login devuelto.

Ambos tipos de autentficiación *digest* y *basic* son utilizadas por los clientes y servidores web, sin embargo, no deberían ser utilizados como un grado de protección para información sensible o accesos seguros. Es común emplear el mismo usuario y contraseña para diferentes servicios, en estos casos habría que tener en cuenta, que los recursos que queremos proteger con este método, no sean recursos muy comprometidos, y que las credenciales no funcionen en otro servicio como puede ser el correo o acceso a información personal.

Autenticación proxy

Las secuencias anteriores son de un cliente pidiendo un recurso protegido a un servidor web. Pero lo mismo se aplicaría cuando un proxy requiere validación para acceder a un recurso. Veamos tambien este caso, y que códigos muestra cuando se trata de un proxy.

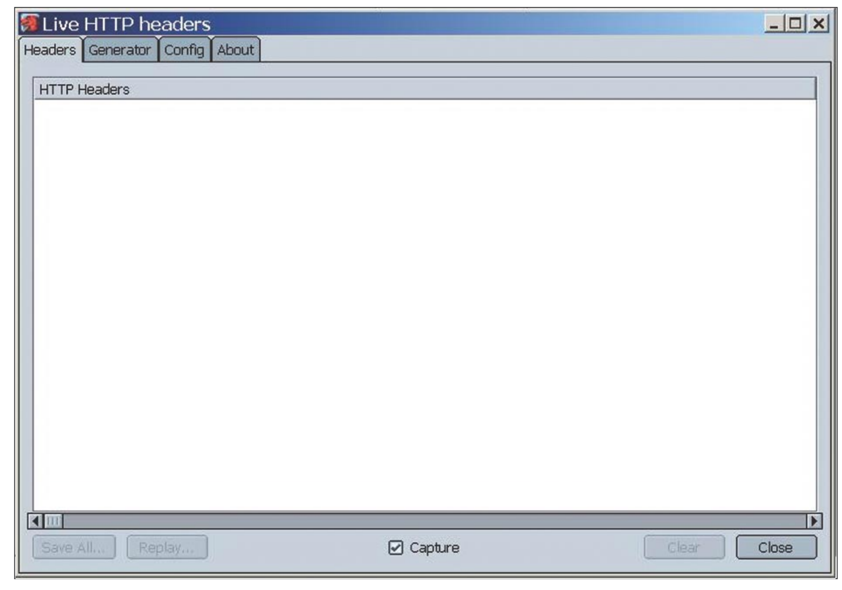


Figura 7. *Recuadro de livehttpheaders*

CURSOS DE VERANO
SEGURIDAD INFORMATICA

Ultima semana de julio – Barcelona
Primera semana de agosto – Gran Canaria



Paseo Montjuic, 20, 5-1
08004 Barcelona
http://www.apif.info

PROGRAMA PREVISTO:

- Taller de migración de Windows a Linux
- Taller de criptografía
- Taller de firewalls
- Taller de configuración de un servidor de correo electrónico gratuito (Hmailserver)
- Taller de trucos contra el spam y el phising
- Taller de capacitación de peritos informáticos
- Taller de securización de redes wireless
- Taller de seguridad en Internet
- Taller de redes
- Taller de Cisco
- Taller de análisis forense informático
- Taller de software libre



- Todos los talleres tienen una duración entorno a las 3 ó 4 horas.
- Precio único: **120€** por taller.
- Oportunidad para aprender algo nuevo y realizar turismo en una ciudad que no se conoce.
- Más información en nuestra web http://www.seguridad0.es
- Llámanos para hoteles recomendados por la zona: 902 900 733

Dirigido a profesionales de la seguridad informática, personal preocupado por la securización de sus estaciones de trabajo y servidores, webmasters, administradores de sistemas, y usuarios finales.

APIF ofrece diversos servicios a centros de formación, consultoras, empresas informáticas, oganizaciones sindicales, patronales y todo aquel que organice cursos de informática. La Asociación de Profesores de Informática Freelance cuenta con más de 1.000 asociados a los que ofrece bolsa de empleo, formación contínua, certificados, ofertas comerciales exclusivas, y asesoría jurídica y laboral.

Seguridad0®, marca comercial de Soluciones Informáticas Seguridad Cero, SL. mayorista y proveedor de soluciones de seguridad informática para PYMES y grandes empresas. Ofrece soluciones innovadoras que abordan los nuevos retos y amenazas a los que se enfrenta cualquier organización en Internet, proveyendo el software necesario y en español, así como cursos de formación online y presenciales. **Seguridad0®**, se ha convertido en un referente de la seguridad informática por su línea de productos. Además, cuenta con la certificación de IQUA como sitio Web que cumple todas las normativas legales en cuanto a la LOPD y LSSI. **Seguridad0®** es ampliamente conocida por la web de noticias online Seguridad0.com, con más de 2.000 visitantes diarios. Entre otros servicios gratuitos ofrece una lista de distribución, Seguridad0.info, con cerca de 1.000 inscritos, y un boletín semanal, Informativos.ws, con más de 5.000 suscriptores.

Un ordenador por alumno. Entrega de programas para el seguimiento del curso en un CD-ROM e información adicional en formato PDF. Incluye herramientas open source y comerciales.



SEGURIDAD0®
c/ Velia, 30-34, 4-1, izqda.
08016 Barcelona.
Tel: 902 900 733
http://www.seguridad0.es



Figura 8. *Prompt de usuario*

Paquetes capturados con ethereal en una conexión solicitando un acceso a internet con un servidor proxy requiriéndonos validación para ello.

Con un servidor proxy configurado en nuestro navegador, realizamos una petición para poder navegar:

```
GET
http://www.google.com/ HTTP/1.1\rn
    Host: www.google.com\rn
    User-Agent: Mozilla/5.0
(Windows; U;
 Windows NT 5.1; en-US; rv:1.7.12)
    Accept:
 application/x-shockwave-flash,
text/xml,application/xml,*/*;q=0.1\rn
    Accept-Language:
 en-us,en;q=0.7,es;q=0.3\rn
    Accept-Encoding: gzip,deflate\rn
    Accept-Charset: ISO-8859-1,
utf-8;q=0.7,*;q=0.7\rn
    Keep-Alive: 300\rn
    Proxy-Connection:
 keep-alive\rn
```

El proxy nos contesta indicándonos que necesitamos validarnos para poder navegar.

```
    HTTP/1.0 407
Proxy Authentication Required\rn
    Server: squid/2.5.STABLE12\rn
    Mime-Version: 1.0\rn
    Date:
 Mon, 16 Jan 2006 13:01:19 GMT\rn
    Content-Type: text/html\rn
    Content-Length: 3283\rn
    Expires:
 Mon, 16 Jan 2006 13:01:19 GMT\rn
    X-Squid-Error: ERR_
CACHE_ACCESS_DENIED 0\rn
    Proxy-Authenticate:
 Basic realm=
""Proxy Authentication
   (user/passwd)""\rn
    X-Cache:
 MISS from proxy.es\rn
    Proxy-Connection:
 keep-alive\rn
```

Entonces nuestro navegador lo interpreta como un desafío/respuesta de autenticación básica y nos muestra el login para introducir los datos requeridos – ver Figura 9.

Algunos navegadores no interpretan bien el *realm* por lo que en algunos si que se verá en el cuadro anterior el mensaje *Proxy Authentication* (*user/passwd*), este no es el caso como se puede ver, pero nos sirve para el ejemplo.

Introducimos usuario y password y nuestro cliente envía los siguientes datos de vuelta al proxy:

```
    GET
http://www.google.com/ HTTP/1.1\rn
    Host: www.google.com\rn
    User-Agent: Mozilla/5.0
(Windows;
U; Windows NT 5.1;
 en-US; rv:1.7.12)
    Accept:
 application/x-shockwave-flash,
text/xml,
image/gif;q=0.2,*/*;q=0.1\rn
    Accept-Language:
 en-us,en;q=0.7,es;q=0.3\rn
    Accept-Encoding:
 gzip,deflate\rn
    Accept-Charset:
 ISO-8859-1,utf-8;q=0.7,*;q=0.7\rn
    Keep-Alive: 300\rn
    Proxy-Connection:
 keep-alive\rn
    Proxy-Authorization:
 Basic
 ZWNhc2JhcOB1bmF2Lm
VzOnBydWViYTAx\rn
```

El servidor proxy comprueba internamente que efectivamente el usuario y password son válidos y nos da acceso al recurso:

```
    HTTP/1.0 200 OK\rn
    Cache-Control: private\rn
    Content-Type: text/html\rn
    Content-Encoding: gzip\rn
    Server: GWS/2.1\rn
    Content-Length: 1408\rn
    Date:
 Mon, 16 Jan 2006 13:05:40 GMT\rn
    X-Cache:
 MISS from filter\rn
    Proxy-Connection:
 keep-alive\rn
```

En lugar de contestar con un código `401` de HTTP, al tratarse de un servidor proxy, muestra el código `407` (autentficación de proxy requerida), y la cabecera que añadía WWW-authenticate el servidor web, ahora al tratarse de un proxy, añade la cabecera *Proxy-Authenticate*. Y todo el proceso sería idéntico que el de un acceso a un recurso web restringido, pero con estas mínimas diferencias.

Visión general de la autenticación HTTP

El esquema de la autenticación básica, no es un método seguro para la autenticación de usuarios, no protege de ninguna manera la identidad del usuario, que es transmitida en texto claro a través de la red. HTTP por otra parte, no evita el que se puedan utilizar esquemas adicionales de autenticación y mecanismos de encriptación empleados para incrementar la seguridad o añadir mejoras en cuanto a la seguridad de la autenticación básica.

A pesar de la debilidad de este tipo de autenticación tal y como hemos visto anteriormente, este método

Tabla 1. *Autenticación de servidor web y autenticación proxy*

Servidor Web	Servidor Proxy
Código de estado sin autorización: 401	Código de estado sin autorización: 407
WWW-authenticate	Proxy-Authenticate
Authorization	Proxy-authorization
Authentication-Info	Proxy-Authentication-Info



Figura 9. *Muestra del login para introducir los datos requeridos*

se usa en diversos entornos donde el mayor peligro reside en aquellos donde existe un Single Sign On, es decir, tus credenciales te sirven para validarte en cualquier recurso de la red donde te encuentras. De esta manera, da igual que se empleen mecanismos de acceso seguro con SSL, conexiones seguras encriptadas a nivel de red (IPSEC), programas con login seguro, etc, con que uno sólo de los recursos implemente esta forma de autenticación, tendríamos acceso inmediato a todos los demás servicios disponibles.

Un campo ideal para implementar y utilizar este tipo de autentificación por ejemplo sería, el acceder a estadísticas de uso de un servidor dado, o acceso a cualquier otro recurso, si pensamos que el acceso ilícito a él, no conlleva un peligro potencial. Unido a esto, las credenciales de acceso tendran que ser proporcionadas por el admin del sitio o mediante un programa generador, nunca tendría que poder el usuario elegirlas, ya que estaríamos en el problema anterior. La gente no acostumbar a utilizar credenciales distintas para los diferentes servicios, si no que las reutilizan.

Resumen

Any service in present use that uses Basic should be switched to Digest as soon as practical, (extraido del RFC 2617: HTTP Authentication: Basic and Digest Access Authentication).

La autenticación básica de HTTP es simple y conveniente, pero no es un método seguro. Se debería usar en casos donde el acceso a información se desea que fuera privado, pero no como un requisito absolutamente necesario, y donde su uso no pueda comprometer la seguridad de otros sistemas.

La gente tiende a utilizar el mismo usuario y password para múltiples propósitos, por lo que aunque su uso se haga dentro de entornos fiables y para acceso a información no sensible, siempre existirá el riesgo de que esas mismas credenciales nos den acceso a servicios mas críticos como correo electrónico, documentos personales, bases de datos.

Con un sniffer de red, y unos cuantos scripts apropiados para interpretar el tráfico capturado, en cuestión de minutos, es posible obtener cientos de pares *usuarios/contraseñas* con el método descrito anteriormente. Con la autenticación HTTP, las contraseñas viajan en claro por la red, y en términos de una conexión, las cabeceras con los passwords, no viajan sólo una vez, sino durante todo el tiempo que dure la conexión y en cada transacción que se haga se vuelven a enviar, esto es por la característica del protocolo HTTP que no conserva estado, y es necesario recordar los datos que se suministran en cada conexión que se hace con el servidor web o proxy.

Para mejorar este método de autenticación, o sustituirlo por otros más seguros sería conveniente:

- Se puede combinar con SSL para reforzar la seguridad encriptando todos los datos de la transmisión.
- Se puede sustituir por la autenticación digest.
- Kerberos
- Eliminarlo donde no sea necesario. ●

En la Red

- Plugin para mozilla.
- *http://livehttpheaders.mozdev.org/*
- Autentifcación HTTP: Autenticación de acceso Basic y digest
- *ftp://ftp.isi.edu/in-notes/rfc2617.txt*
- Codificación de transferencia de contenido en Base64
- *http://www.faqs.org/rfcs/rfc2045.html*
- Configurar apache para requerir autenticación.
- *http://httpd.apache.org/docs/1.3/howto/auth.html*
- RFC 2617
- *http://rfc.sunsite.dk/rfc/rfc2617.html*

Ingeniería social

Tomasz Trejderowski 

Grado de dificultad



Una vez alguien ha llamado acertadamente a la ingeniería social, como el hecho de forzar la mente. Es una media aritmética de sociotécnica (ejercer influencia y manipular a las personas) y cracking (violaciones de los sistemas informáticos). Estos dos mecanismos unidos forman una herramienta potente, cuya fuerza de destrucción es desconocida para muchas personas.

Albert Einstein una vez dijo: *Hay dos cosas infinitas: el Universo y la estupidez humana,y no estoy tan seguro de la primera*. Muchas personas asocian esta frase con la sociotécnica e ingeniería social. En cuanto en el primer caso, puede ser que sea justificado, tanto – en relación con la ingeniería social, sería mucho más escéptico. Lo que aquí juega el papel principal es, más bien, la ignorancia, que realmente la estupidez. La falta de conocimiento de ciertas reglas es la base de éxito de tantos ataques sociotécnicos a las empresas. En cambio, se puede llamar el comportamiento de los empleados una estupidez, sólo entonces, si aceptaran abierta y conscientemente los métodos utilizados por un sociotécnico, que atacaría su empresa.

En la ignorancia reside la potencia de la ingeniería social. Si un cracker hábil realiza un ataque contra la seguridad de una red empresarial, siempre aparecerá alguien, quien querrá explicar esta situación indicando la ignorancia o falta de competencias de administradores o usuarios de la red. No obstante cuando una empresa sea atacada por un sociotécnico – las victimas de su ataque no sólo no serán conscientes de que están manipuladas, sino también probablemente, incluso después de mucho tiempo desde este evento, no asociarán ciertos hechos al mismo. Aunque suene pérfidamente – el ataque sociotécnico es sutil. Kevin Mitnick – el sociotécnico (y no el cracker – como le llaman los medios) más conocido del mundo repetía siempre en las entrevistas, que muchas veces adquiría la información sólo por eso, que las pidió a alguien. Evidentemente de modo adecuado.

Sociotécnica

La base teórica de la ingeniería social es la sociotécnica, o sea la ciencia de ejercer la influencia, persuasión, y manipulación de la gente. En cada ataque que utiliza la ingeniería social podemos encontrar las huellas de las reglas básicas, que constituyen los fundamentos de la sociotécnica:

- *La regla de la reciprocidad* – cada cosa positiva (la ayuda, el apoyo, un regalo) que recibimos de otra persona genera en nosotros un inmediato e irresistible deseo de reciprocidad.
- *La regla de la prueba social – ¡10 mil clientes no pueden estar equivocados!*. Según

esta regla es más fácil persuadirnos a algo, si se nos demuestra, que otros piensan lo mismo o actúan del mismo modo.

- *La regla de la simpatía* – si alguien nos gusta o parece ser simpático, entonces estamos más dispuestos a realizar sus peticiones.
- *La regla de autoridad* – no tenemos el valor de oponerse a alguien más sabio, con mayor experiencia o *en una posición superior*, que nosotros. Este mecanismo funciona incluso, si estamos convencidos de que la decisión tomada por esta persona o su actitud son incorrectas.
- *La regla de comprometimiento y consecuencia* – si nos comprometemos en algo, aspiraremos consecuentemente a realizar el objetivo intentado.
- *La regla de inaccesibilidad* – la percepción del valor de una cosa aumenta en nuestra opinión, si es temporal o prolongadamente inaccesible. Existe también *La regla de inaccesibilidad inversa* – la cosas frecuentes, obvias y de acceso fácil tienen según nosotros un valor pequeño.
- *La regla de valor y provecho* – vale la pena luchar por las cosas de valor (tanto las cosas materiales, como por ejemplo dignidad, honor, buena reputación, fama). Si un sociotécnico causa la impresión de que estos aspectos de nuestra vida están amenazados, puede manipularnos fácilmente para que actuemos de forma involuntaria y extraordinaria.

Estas reglas se utilizan con frecuencia en la manipulación sociotécnica (en los medios de comunicación, en la política, en la vida profesional). Si hablamos de la ingeniería social, más bien se utiliza varias combinaciones de estas reglas.

No obstante hay que recordar, que el mecanismo más popular utilizado por los sociotécnicos en sus ataques a las empresas, es simplemente la mentira.

Falta de términos claros

Una parte de personas considera necesario separar claramente los dos términos. La influencia totalitaria en todas las naciones (grandes grupos sociales) es la *ingeniería social de comunidades*. La manipulación informativa de un individuo (pequeño grupo de personas) es la *ingeniería social*. En el segundo caso se utiliza también los términos *socio-informática* y *socio-hacking*.

De otro lado muchas personas creen, que informática y totalitarismo (y además: marketing, medios de comunicación, negocios, etc.), son dos campos de sabiduría muy lejanos. A pesar de eso, se puede tratar de influenciar en la gente en estos campos, con unos mecanismos iguales e idénticos. Por eso, en todos los casos, utilizan el mismo término – *ingeniería social de comunidades*.

El término más popular parece ser: *ingeniería social* y ha sido usado en este texto, aunque tiene el mismo número de partidarios, que los enemigos.

El nivel de seguridad

La conciencia del peligro, que lleva la sociotécnica e la ingeniería social, sigue siendo menospreciada y marginada en la mayoría de las empresas.

¿De dónde saco unos datos tan sensacionalistas? Presentaré mi propio ejemplo, que, según mi opinión, demuestra muy bien la escala del problema. Pese a escribir los artículos, soy también profesor. El centro de formación, donde trabajo, es el único en la región, que organiza los cursos de sociotécnica e ingeniería social. Durante el año organizamos dos cursos y tenemos problemas para captar *diez* clientes. Y vivo y trabajo en una ciudad (Katowice, Polonia), que tiene trecientos mil habitantes, una zona exterior donde viven mas de 2 millones de personas y unas 50-80 mil empresas (una situación parecida tiene lugar en muchas de las ciudades europeas). Es probablemente un testimonio suficiente, de que la conciencia del peligro en este campo es casi abstractamente baja.

Un sociotécnico cuando ataca, daña siempre el vínculo más débil de cada empresa: el hombre. La sociotécnica y la ingeniería social atacan al nivel del subconsciente de la mente humana – son las esferas de la actuación, de las cuales muchas veces no nos damos cuenta; los mecanismos instintivos y automáticos. Ni los contrafuegos, ni los programas antivirus, ni incluso unos cientos de miles de euro gastados en la seguridad informática, nada de eso nos protegerán contra tales ataques. La única protección contra eso, es la formación constante, los cursos de sociotécnica, que eliminarán la ignorancia, la cual he mencionado al principio del artículo.

Como demostraré a continuación, sólo aparentemente el peligro no nos concierne personalmente, cuando la realidad es que está acechando a cada empresa, en cada ciudad y en cada momento.

¿Quién y porqué?

No se puede determinar claramente quien usó la sociotécnica por primera vez. Ya hace miles de años, los faraones de Egipto utilizaron los momentos de eclipse solar calculados matemáticamente para manipular su pueblo, hacer creer una visiones espantosas y recibir la promesa de obediencia.

En cuanto a la ingeniería social, o sea el uso de la sociotécnica para los ataques a las empresas y los sistemas informáticos, la cuestión del primer uso de este mecanismo es también muy difícil de averiguar. Muchas personas dicen, que el primero fue el famoso Kevin Mitnick. No obstante en mi opinión la verdad es que solamente popularizó esta técnica de adquirir informacion. Probablemente la ingeniería social existe, desde que se usan los ordenadores. O sea desde los años sesenta o incluso cincuenta del siglo pasado.

La respuesta a la pregunta, *¿porqué se utiliza este mecanismo?* es banal y ya lo he explicado en el fragmento anterior. Es simplemente muy rentable. En Europa todavía (afortuna-

damente) domina la primera fase de desarrollo, durante la cual la mayoría de los ataques están realizados por unos sociotécnicos de formación autodidacta. Lo hacen generalmente para resolver sus pequeños problemas particulares, que a veces son difíciles de arreglar por la vía *tradicional*. Muchas veces utilizan la ingeniería social sólo para probarse a sí mismos, que están capacitados para ello (*¿Sociotécnicos de sombrero blanco?*). Pero esta época no durará mucho. El momento en que los sociotécnicos serán empleados por la competencia, para robar los secretos comerciales o llevar una empresa a la bancarrota – el modelo norteamericano – no están lejanos. Es cuestión de unos cuantos años.

La cuestión de la reputación

Sobre todo hay que preguntarse, ¿puede un empleado o el propietario de la empresa asociar determinados eventos (como frecuentemente la perdida de datos confidenciales) con la actividad de un sociotécnico? Son unos ataques sutiles (lo que significa – difíciles de notar) y no dejan daños que sean visibles inmediatamente (como por ejemplo el disco duro formateado, la página web cambiada – el efecto de la actividad de un cracker). Por eso las posibilidades de descubrirlos (incluso después de muchas semanas) por el personal no cualificado y no muy consciente del tema – son, según mi opinión, mínimas.

Sin embargo, si la empresa descubre, que ha sido atacada por un sociotécnico ¿lo admitirá entonces? No. Ya que no consigue nada haciendo esto, y puede perder mucho – sobre todo la confianza de sus clientes y una buena imagen de empresa.

Por eso, aunque seguramente existen algunos datos estadísticos sobre el tema, su verosimilitud hay que evaluarla con cautela. ¿Cómo es posible, que alguien de alguna manera consiga datos verosímiles sobre el tema, si las informaciones de este tipo son el mayor secreto de cada empresa? Un ataque sociotécnico, pese a las perdidas financieras calculables, es también un reconocimiento de la propia debilidad. Algo intolerable.

```
%7E%E9*%13%88%C9%DA%1D%EEve%F3
%E4%A5%15%E8P%5E%E7%F7%AF%5E%D
Bsa%B9%A8%89%16%CA%15%2F%82%DE
%E4v%DB%1D%7C%AD%D9%E0%1F%B8%0
1O%F9%D9%D0%06%B7%5Ei%DA%81%9D
2%A0n%10%F7%D9%8D%09%DCJj%BE%C
4%B9%26%E0%3E%1E%CC%D8%D8*%83%
5Bs%B3%BC%91P%EB-%2B%B2%C1%E4b
%EA%60%00%86%E5%D6y%BA%0Ca%E4%
B6%9Fq%8A%0A%27%DF%ED%C51%9AEA
%93%81%92%2C%EF%10G%DE%F1%ACb%
CE4i%C7%F8%C7%0E%A9%3E4%A4%91%
E8a%CE%7Dh%93%B6%EA%0A%E0%5Ct%
D1%BC%C8Z%BB%7D%7F%9F%D0%FE%0C
%A9BS%DD%F6%88%21%AAhU%98%DB%F
```

91P%EB-%2B

Figura 1. *¿Quién de las personas ajenas descubrirá el lema "escondido" en tal impreso?*

Una espada de dos filos

Conociendo las técnicas de ingeniería social y los métodos de prevención, con frecuencia es muy difícil resistir la voluntad de usarlos para nuestros propios fines. La evaluación de la moralidad y las posibilidades de descubrir tal actividad, los dejo en manos de los Lectores. Quiero solamente señalar, que la ingeniería social, como cada forma de manipulación sobre las personas, sin su voluntad y conciencia, para obtener unos beneficios propios, es en Polonia (y leyes similares existen en casi todos los países) un crimen. Párrafo 1 de artículo 267 del *Código Penal* dice: *Él que sin estar autorizado obtiene la información que no esta destinada a él, abriendo una carta cerrada, interceptando las informaciones mediante la conexión a un cable de transmisión o violando la protección electrónica, magnética o otras protecciones particulares, será castigado con la pena de limitación de libertad o pena de prisión de 2 años como máximo.*

La falta de verificación

Si uno entiende la idea de la herramienta más simple y más popular de cada sociotécnico – la mentira – entonces debería encontrar automáticamente la solución más simple. Es la verificación. Si existe la posibilidad de hacerse pasar por otra persona, hay que verificarlo, verificarlo y una vez más verificarlo – a cada paso. Prácticamente cada conversación en la empresa, que concierne los aspectos estratégicos (información confidencial, seguridad, dinero) debería realizarse a base de devolver la llamada. Si al empleado Pepe Pérez llama el administrador Juan Nadie, Pepe Pérez cuelga el teléfono y vuelve a llamar a Juan Nadie (por supuesto utilizando el número de teléfono que conoce y no el teléfono dado por Sr. Nadie), para asegurarse en el cien por cien, que le ha llamado verdaderamente Juan Nadie y no un sociotécnico, que se hace pasar por él.

Es la teoría. En la práctica las empresas están todavía lejos de la posibilidad de implementar tal solución. El problema consiste en eso, que no en todos los casos es cuestion de dinero. Sabiendo que las llamadas en la interna red telefónica empresarial son gratuitas, el coste de implementar tal solución, no debería ser muy alto. No obstante muy frecuentemente las personas responsables piensan, que los medios de seguridad tan avanzados no son necesarios. Desgraciadamente es un pensamiento muy peligroso.

¿Cuando una información inocente puede hacer daño?

¡Siempre! Cada información es peligrosa. Incluso la más simple – como el nombre y apellido de una compañera, que está sentada en el lado opuesto del escritorio. Ya que, incluso si eso no le ayuda al sociotécnico directamente, lo puede utilizar para verificar su identidad artificial, creada durante una conversación con otra persona. Lo voy a repetir otra vez. Cada información, incluso la más banal – que sale fuera de la empresa, es peligrosa.



Llegamos a una simple conclusión. Uno no se puede defender contra la sociotécnica y la ingeniería social. No existe, nunca ha existido y nunca existirá la manera eficaz de defenderse. Se puede, como máximo, probar a disminuir su actividad y reducir el riesgo al mínimo. No se puede eliminar por completo.

Hacer la telaraña

La mayor ventaja de un sociotécnico y al mismo tiempo el mayor peligro a las empresas es que el atacante tiene el tiempo y la paciencia. Un ataque sociotécnico no se realiza sin cuidado, por un diletante, que ha leído la traducción (generalmente fatal) de un libro norteamericano sobre ese tema y quiere comprobar sus capacidades. Es un juego estratégico preparado con arte, elaborado en muchos frentes, que contiene las soluciones alternativas y unos escenarios de las situaciones atípicas.

El mecanismo de tal ataque es por lo general el mismo o muy parecido. Se parece al hecho de hacer la telaraña. De unos hilos pequeños, que por sí solas no son el obstáculo ni siquiera para una mosca pequeña, con el tiempo se crea una red excelente, que encerrará no sólo una banda de moscas, sino también un mosquito grande y muchos otros insectos.

Para un sociotécnico no existen las informaciones sin valor. Cada una, incluso la más pequeña le puede ayudar. Estas, que aparentemente parecen ser las menos importantes, son para él las más preciosas. Es por eso, que son tan fáciles de obtener – ya que nadie les trata como secretos y no se las protege especialmente – y pueden ser usadas para autentificar las informaciones preparadas, y en efecto obtener unos datos de valor.

El algoritmo de una actuación por ejemplo, uno de tantos, podría ser el siguiente:

- obtener el nombre y apellido de cualquier empleado de la empresa;
- el nombre y apellido sin problemas revela el departamento, donde trabaja esta persona;
- el hecho de obtener la dirección de correo electrónico de un empleado, muchas veces ni siquiera exige conocer el departamento, donde trabaja – estos datos se publican con frecuencia en la página web de la empresa;
- se puede llamar a este empleado, fingir al administrador y gracias a una simple manipulación conseguir su login y contraseña para la red empresarial. Sin embargo esto no es el objetivo como tal;
- teniendo el e-mail, uno no debería tener mayores problemas con el hecho de conseguir el número interno de este empleado (por ejemplo en la lista de salarios, etc.).

Poseyendo el nombre, apellido, departamento, e-mail, número de identificación y muchas veces también el login y la contraseña tenemos las bases para robar la identidad. Un sociotécnico equipado con estas informaciones puede tranquilamente y con éxito pasarse por un empleado de la empresa y continuar su ataque sociotécnico en la dirección deseada.

Por favor, noten no sólo el contenido, sino también la idea del plan de actuación. En cada caso un sociotécnico llama a otra persona, se hace pasar por alguien diferente (el administrador, un empleado de la empresa, un cliente), presenta una historia falsa, pero con los detalles que le autentifican y en modo muy ingenioso pide una información simple, que no significa nada.

Cada uno de los empleados "atacados" consentirá en eso y no resiste demasiado, ya que asume conscientemente (y tiene razón), de que no revela ningunos secretos comerciales o empresariales, sino sólo *unas informaciones que no significan nada*.

Métodos para adquirir información

Debido al tamaño y el carácter de esta publicación, estoy forzado a escoger y dar como ejemplos solamente los métodos de ataques más importantes.

- Utilizando la regla de inaccesibilidad (curiosidad), un sociotécnico puede manipular a un empleado de la empresa para que haga algo atípico, por ejemplo enviando un e-mail o poniendo a su alcance un disquete que contiene supuestamente unos datos secretos (el sistema de premios, los salarios de la junta directiva, unos detalles picantes de su vida privada etc.). Al abrir el documento adjunto (al iniciar el programa), en el ordenador de la victima se instala un troyano, que registrará las teclas

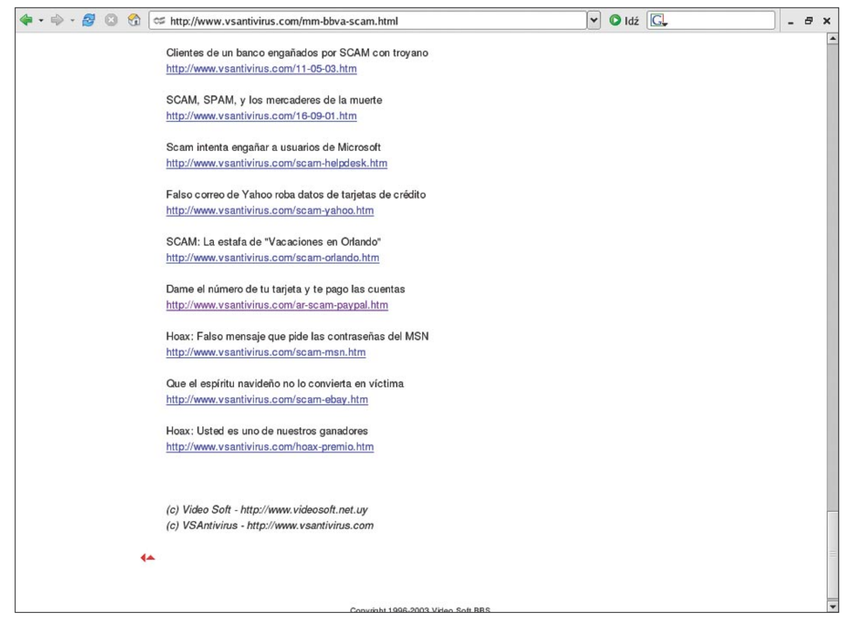


Figura 2. *Ejemplos de ataques sociotécnicos*


http://www.laflecha.net/perfiles/tecnologia/kevin_mitnick/

LaFlecha
Tu diario de ciencia y tecnología

Web LaFlecha Buscar

PORTADA

CANALES
BlackHats
Ciencia
Comunicación
Curiosidades
e-Administración
Empresas
Eventos
Seguridad
Software Libre
Telefonía
Videojuegos
Wireless

EL PERIÓDICO
Editorial
Perfiles
Blogs

SERVICIOS
Boletines
Contactos
Formación
Viviendas
Tienda

COMUNIDAD
Emails de los lectores

FORMACIÓN
Cursos sobre:
Buscar

Perfiles / Tecnologia / Kevin Mitnick

PERFIL

EL CHACAL DE LA RED

Kevin Mitnick

Esta es su historia: Kevin David Mitnik, alias "El Cóndor", nació el 6 de agosto de 1963 en Van Nuys, un pequeño pueblo al sur de California. Hace 41 años. Kevin no tuvo una infancia fácil. Sus padres se divorciaron, y el entorno en el que creció supuso una de las causas que le llevaron a encerrarse en casa y permanecer en soledad, momento (1970s) en el que fue seducido por las mieles de la informática; en concreto, por el poder que entrevio que podía obtener a través de las redes telefónicas. De hecho, a pesar de que Kevin ha recibido calificativos tan diversos como hacker, cracker, pirata informático y otras palabras nada agradables que eludo mencionar, lo cierto es que en esencia, Mitnik siempre ha sido un Phreaker; el mejor phreaker de la historia, según muchos.

18/01/2005 | por Sarah Romero

Desde muy joven sintió curiosidad por los sistemas de comunicación electrónica, cultivando y desarrollando un deseo cuasi obsesivo por investigar, aprender y lograr objetivos aparentemente imposibles. En la adolescencia Mitnick ya se había convertido en todo un fenómeno. A los 16 años marca su primera línea tras la barrera del "lado oscuro", del que ya nunca se separaría: se saltó la seguridad del sistema administrativo de su colegio; sin embargo, no lo hizo para alterar las calificaciones escolares, como podría pensarse, sino "solo para mirar".

De aquí en adelante comenzaría su andadura como delincuente. La fecha, 1981. Kevin y dos amigos suyos irrumpieron en las oficinas de Cosmos (Computer System for Mainframe Operations) de la compañía Pacific Bell – que era una base de datos utilizada por la mayor parte de las compañías telefónicas norteamericanas para controlar el registro de llamadas -. Una vez allí obtuvieron la lista de claves de seguridad, la combinación de las puertas de acceso de varias sucursales, y manuales del sistema COSMOS, entre otras cosas. De hecho, se comentó que lo sustraido por Mitnick y sus amigos tenía un valor equivalente a 170.000 euros.

PUBLICIDAD
Anuncios Goooooogle

Dominios
Registro y alojamiento de dominios Empresa profesional
dominios.net

Tellink Sistemas S.L.
Tarificadores para Locutorios Gestión de cibersalas
www.tellinksistemas.com

Llame al 11882
Servicio de información telefónica Atención personalizada y eficiente
www.guias11882.com

Telefonia por internet
Telefonia por internet en Guatemala Precios unicos en el mercado
www.nexo.com.gt

Rebelio

Figura 3. *Información sobre Kevin Mitnick*

pulsadas y le dará al atacante el acceso a la red empresarial y a los datos guardados en el disco duro

- Utilizando la regla de autoridad, el atacante puede fingir al superior y forzar directamente a un subordinado a revelar unas informaciones. Consiguiendo el nombre y apellido de la asistente del presidente, es posible llamar luego a otra persona y decirla, que llamamos a petición de ella. En la empresa casi nadie rechazará la petición de los miembros de la junta directiva y de sus colaboradores más próximos.
- La regla de simpatía puede ser utilizada en una imaginada situación de peligro de una persona conocida y amada. Un sociotécnico puede llamar a una empresa, pasarse por un empleado de banco y decir, que un compañero o un colaborador de la persona, a la cual está llamando, a causa de... (una mentira verosímil), no recibirá su salario a tiempo, pero nosotros le podemos ayudar, dando... (una información casi sin importancia, pero generalmente no accesible – por ejemplo el identificador de este empleado). En una situación así, dejándose llevar por la simpatía (que se está utilizando aquí) y las ganas de ayudar, revelamos fácilmente unas informaciones, que por lo general no deberían salir de la empresa.
- Una solución alternativa, a lo mencionado arriba, es llamar a la persona en cuestión y convencerle (manipulando con el miedo, un aspecto de la regla de valor), de que la transferencia de su salario ha salido a un banco inadecuado. Un empleado agitado (emociones) es muy fácil de manipular. En esta situación es posible sacar prácticamente casi toda la información, para ayudar según le parece.
- La cuestión de la ayuda se utiliza en la sociotécnica inversa. Un sociotécnico puede organizar cierta situación (por ejemplo la falta de acceso al Internet en un periodo muy *caliente* para el empleado), antes fingiendo al administrador y dando el número de una *línea de emergencia* en caso de problemas. Si la víctima (a causa de una mistificación ingeniosa) llama al sociotécnico por sí sola, es casi seguro que apagará todos los posibles mecanismos de defensa y se dejará manipular fácilmente.
- El mecanismo de pedir la ayuda o las ganas de ayudar es uno de los más populares entre los sociotécnicos. Por eso hay que tener mucho cuidado a quien ayudamos y de quien son las informaciones que utilizamos. El hecho de ayudar inicia el mecanismo de la regla de la reciprocidad, lo que puede hacer el ataque de un sociotécnico eficaz.
- En casos extremos un sociotécnico no tiene que hacer absolutamente nada, ya que los empleados de empresas no cualificados frecuentemente solos revelan las informaciones. Por ejemplo en el protocolo, que termina el contrato con el operador de telefonía móvil – el cliente recibe una copia de este protocolo – está imprimido el número SFID del empleado que finaliza el contrato. Es un número confidencial, interno, que funciona en la red y claramente identifica al empleado. Las informaciones de este tipo absolutamente no deberían encontrarse en los documentos de libre acceso, imprimidos para los clientes. En uno de los ambulatorios de Katowice (ciudad en Polonia) se produjo un descuido mucho peor. Sobre el escritorio en la recepción estaban los documentos de nuevos pacientes, que todos podían ver. Las informaciones destinadas a *NFZ* (*Narodowy Fundusz Zdrowia – Fondo Nacional de Salud de Polonia*) que contenían todos los datos, es decir el nombre, apellido, la dirección, el teléfono, NIF, REGON – número de registro económico nacional. Una persona ingeniosa, no necesita nada más, para robar la identidad.
- Para conseguir el número de teléfono móvil de un empleado, conociendo solamente su nombre y apellido, basta con usar un truco fácil. Un sociotécnico puede llamar a la recepción (sede) de la empresa, antes asegurándose, de que en este momento la persona está fuera de la oficina (vacaciones o ausencia temporal). Finge ser el administrador, inventa una historia (por ejemplo el balbuceo técnico incomprensible) y pide que se deje en el escritorio del empleado una nota: *Dame un toque en el número*

XXX. ¡Es muy urgente! Administrador. Si el empleado vuelve a llamar – el sociotécnico tiene su número (incluso no tiene que contestar). Si el número es oculto, es suficiente pedir que en la nota se ponga: *Mándame un SMS en el número XXX, no puedo contestar a las llamadas*. En casos extremos incluso uno no tendrá que esperar, ya que la recepcionista asustada y adecuadamente manipulada le dará por sí sola el número de móvil del empleado.

- Basura – los papeles echados por la empresa contienen unas centenas de datos (las direcciones e-mail, identificadores, teléfonos internos, nombres de los superiores etc.) de los empleados. Y se los destruye raramente. Un sociotécnico de este modo no conseguirá las informaciones confidenciales directamente (aunque no se puede asumir siempre como cierto), ya que generalmente los datos confidenciales en realidad van a una trituradora o están almacenados y no echados a la basura. Sin embargo los datos *poco importantes* de los residuos no destruidos, sirven perfectamente para *hacer la telaraña* y para que un sociotécnico cree la imagen de una persona, que verdaderamente no existe. Kevin Mitnick en muchas entrevistas mencionaba, que explorando los almacenes de basura en las empresas o incluso sobornando a los empleados de las empresas de residuos, para que le suministran la basura de una empresa dada, era uno de los mejores métodos de adquirir unas informaciones indispensables.
- A nuevos empleados en las empresas en general se las otorga mucha tolerancia e indulgencia. El modelo norteamericano hace uso de un mecanismo, según el cual la competencia manda a su empleado a otra empresa, para que se instale allí para poco tiempo (unos cuantos meses), para que cometa los máximos errores posibles, y si llega el caso, obtenga las máxima información posible. Indulgencia y tolerancia sí, pero al mismo tiempo doble cuidado.
- Un sociotécnico puede llamar a una empresa y causar una situación que exige, que un empleado se marche del teléfono, contando además con que el empleado no va a bloquear el micrófono. Y en muchos casos no estaremos equivocado. Una vez utilizando los servicios de una de las empresas de mensajería más grandes, llamé para averiguar el estatus de mi envío. La verificación por excepción duró unos cuantos minutos, y durante todo este tiempo, cuando mi interlocutor estaba ausente, podía escuchar todas las conversaciones de alrededor – entre los empleados o entre un empleado y un cliente. Oí muchos detalles (las direcciones, los parámetros, los identificadores) que concernían a otros envíos y otros clientes. Es una situación intolerable.
- Muchas veces los administradores de las redes empresariales mismos le facilitan a un sociotécnico su actividad. Les obligan (demasiado) frecuentemente a sus empleados a cambiar la contraseña de acceso por una (demasiado) difícil. El cerebro humano tiene sus limitaciones y el hecho de exigir que un empleado recuerde una larga serie de letras y cifras es una simple exageración. No es extraño que luego utilicen algo más fácil – por ejemplo una simple combinación *de teclado* (*qwe123, zaq12wsx*), cuyas listas están accesibles en muchos sitios en el Internet. O simplemente anotan las contraseñas en las tarjetitas. En cambio si ya tenemos que anotar la contraseña en la pieza de papel, hagamoslo de forma ingeniosa. Es muy difícil recordar veinte cifras y letras. Pero es muy fácil *rodearlas* al principio y al final por ejemplo con cinco signos, no importa cuales, y luego antes y después de un trozo así añadir (o imprimir) diez más, igualmente al azar. Dentro de un bloque tan *denso* de letras, signos y cifras, nadie ajeno será capaz de encontrar nuestra contraseña. Y a nosotros nos basta sólo un vistazo para recordar, de que hay que rechazar las diez líneas de arriba y de abajo y cinco signos de cada lado y simplemente copiar lo que queda.
- Un sociotécnico puede pasarse por un empleado de una empresa más grande (un consorcio grande) que colabora con la empresa atacada. Puede decir que llama del departamento de servicios y hace una simple encuesta para mejorar la cooperación de ambas empresas. Utilizando el mecanismo poco importante-importante, puede meter dentro de unas preguntas banales y evidentes el contenido, que le deje acceder a unas informaciones de valor, facilitadas por un empleado manipulado de este modo. Y teniendo esto, puede a continuación llamar a la empresa real, cuyo empleado fingía antes
- Muchas veces el ataque directo (en las facilidades de la empresa) para conseguir un objeto concreto es imposible o muy difícil de realizar. Ya que todos se conocen y están vigentes unos procedimientos desconocidos a un sociotécnico, y por eso podría ser calado fácilmente. En este caso, disponiendo de una adecuada cantidad de informaciones recogidas antes, se puede fingir a un empleado de la empresa (telefónicamente), y a continuación inventar una

Sobre el autor

Tomasz Trejderowski – metalurgista de formación, escritor, profesor, programador y web diseñador de profesión. Autor de los libros y artículos. Realiza doctorado en la Universidad Politécnica de Silesia.
Página web: *http://www.tomasz.trejderowski.com/*

historia ingeniosa, explicando porque no puede recoger un envío dado personalmente. El atacante puede pedir que se lo deje en la portería e informar que mandará un mensajero/un recadero/un compañero que justo estará cerca etc., para que lo recoja. Luego irá a la empresa sólo, donde ya puede actuar como un hombre *normal*. Nadie le controlará, ya que el verdadero empleado ha avisado, que no puede recoger el envío personalmente. En tal situación es muy fácil de conseguir informaciones imprimidas, muy difíciles de conseguir en otra manera. Sólo hay que tener cuidado, de que esta persona, por la cual se hace pasar el sociotécnico, en el momento esté ausente en la oficina y de que sea imposible contactarla. Para un atacante bien pagado no es un problema grande.

- Si el sociotécnico ataca una de las filiales de una empresa grande (una corporación), es relativamente fácil de fingir a un empleado de otra filial, mentir, de que a causa de una avería, no hay acceso a la red empresarial y así adquirir informaciones que por lo general no están destinadas a alguien *de fuera*.
- Como se ha mencionado en principio, la regla de inaccesibilidad dice, que cuanto más inaccesible es alguna cosa, tanto más sube su valor según el atacado. Un sociotécnico puede crear una situación cuando el acceso a unos datos muy urgentes está limitado (por ejemplo la supuesta *caída* del servidor) y así manipular a un empleado de que le facilite unos datos confidenciales (el login, la contraseña) para que el acceso a estos datos sea recuperado, según parece, de forma más rápida.
- Unos mecanismos muy semejantes para provocar el sentimiento de peligro (*Su tarjeta de crédito será desactivada, por favor paseme su número y el código, para anular el proceso de la desactivación.*) u ofrecer una promoción (*Pasanos tu login y contraseña a eBay, para obtener 5 euros en la cuenta de usuario.*) son muy populares y utilizados frecuentemente por los phisers. El mecanismo es muy simple y consiste en persuadir a la víctima para que revele las informaciones necesarias para realizar un fraude. De ningún modo es necesario ser un cracker. Son suficientes: un poco de ingenio, un poco de ingeniería social y un poco de credulidad (y muchas veces – la codicia) de las víctimas.

¿Cómo defendernos?

Son sólo unos ejemplos. Más de diez. Un sociotécnico profesional, ganándose la vida con su actividad ilegal, conoce unas centenas o unos miles de posibilidades. Y observando intensamente la vida y las relaciones humanas es capaz de inventar cada día al menos una más. Sus ventajas principales son la mentira, la seguridad en sí mismo, la paciencia y la experiencia.

No existe una manera segura de defenderse contra un sociotécnico y probablemente no existirá nunca. Si grabamos en la mente ciertas reglas de actuación, esto puede hacer la vida de muchos de los sociotécnicos más difícil e imposibilitar el hecho de llevar a cabo muchos de los ataques sociotécnicos. Estas reglas son:

- Verificar, utilizando todos los medios, si la persona, con quien hablamos, es esta, por la cual se hace pasar. Un sociotécnico por medio del método de hacer la telaraña puede conseguir muchas informaciones sobre la empresa y la víctima, pero es casi seguro de que no será capaz de conseguir a todas. A la empresa le ayuda mucho y al mismo tiempo al sociotécnico le complica la vida, si no se violan ciertas reglas de seguridad establecidas, solamente por la razón, de que él nos pide esto. Si para la verificación sirven seis códigos y si al azar escogemos un C, deseemos de que el interlocutor pase el código C. No obedezcamos a su petición de cambiar de código, solamente por eso de que supuestamente alguien está sentado delante de su ordenador. Esto puede ser una mentira ingeniosa y el sociotécnico, que nos está llamando puede simplemente conocer sólo un código e intentar manipularnos en esta manera.
- No emocionarse, no dejarse influir por la presión de la autoridad y el miedo;
- No abrir los archivos adjuntos a los e-mails y los archivos en los disquetes y discos de las fuentes desconocidas;
- Cuidar de que la basura y otros materiales impresos que salen de la empresa sean destruidos regularmente y minuciosamente. No revelar unos datos confidenciales en los documentos de libre acceso. Bloquear las llamadas telefónicas, para que una persona ajena no pueda oír lo que pasa en la oficina;
- Controlar muy detalladamente a quien ayudamos y quien nos ayuda a nosotros;
- Realizar la formación regular del personal en cuanto a la sociotécnica e la ingeniería social.

Resumen

Sociotécnica es el hecho de manipular el comportamiento de la gente, gracias a que es posible obtener beneficios o unos datos confidenciales. La sociotécnica existe desde el principio de la historia de la humanidad, no obstante el uso de sus reglas para los ataques a las empresas y los sistemas informáticos empezó desde hace varias decenas de años.

Espero, que haya logrado a alegar razones lo suficientemente fuertes, para convencer al menos unas cuantas personas, de que el problema no sólo existe, no sólo es un peligro real, sino también, sobre todo – concierne prácticamente a cada empresa, cada persona y cada aspecto de la vida: tanto profesional, como privada. ●



Entrevista

Nunca te confies, no estamos completamente seguros

Entrevista a Dr. Lars Packschiese

Hablamos con nuestro invitado, doctor Lars Packschiese, científico sobre las aplicaciones criptográficas y administrador de aplicaciones y de protección de datos en el entorno Linux, SunOS/Solaris, IRX y AIX del Centro Regional de Contabilidad (Centro Regional de Contabilidad, RRZ) de la Universidad de Köln. El Dr. Packschiese es autor del libro „Criptografía práctica en Linux".

H9: Trabajas como científico en el Centro Regional de Contabilidad de la Universidad de Köln; publicaste también el libro *Criptografía práctica en Linux*. ¿Crees que esto es suficiente para que los usuarios de RRZ se sientan seguros?

LP: Sí, podemos decir que la mayoría de los usuarios de mi entorno es consciente de los problemas habituales, sobre todo, de los relacionados con la comunicación por correo electrónico. Sin embargo, en muchas conversaciones observo que la mayoría de ellos quiere emplear el cifrado, pero no sabe por donde empezar. Algunos usuarios admiten abiertamente que son demasiado perezosos para aplicar la clave GPG, a la cual tienen acceso. Otros, quisieran cifrarse, pero no pueden convencer a la otra parte. En cuanto a SSH en mi entorno, la cosa es totalmente diferente. Por ejemplo, los usuarios de los ordenadores de alto rendimiento tienen a su disposición SSH solamente si se registran adecuadamente en estos ordenadores.

H9: ¿Cuál es en tu opinión la causa del escaso empleo de la clave GPG en la práctica?

LP: Sobre todo que la gente no sabe cómo empezar. Si alguien se atreve a dar el primer paso y ha genera unas claves o un certificado, se encuentra ante el siguiente obstáculo – emplearlo en la práctica.

H9: ¿Es posible estimar aproximadamente el porcentaje de los usuarios que protegen sus correos o bien que al menos aplican las reglas mas elementales de seguridad?

LP: Las conclusiones del Ostermann Research (*http://www.ostermanresearch.com*) del año 2004 demuestran que el 20 por ciento de los empleados de grandes empresas son *usuarios frecuentes* que cifran los correos, siempre que la empresa aporte soluciones de cifrado. No sé exactamente cómo se divide entre OpenPGP, S/MIME y otras técnicas. En universidades yo estimo que hay más usuarios de este tipo.

H9: ¿Cuántos usuarios de tu entorno piensas que hay que sean susceptibles con este tema?

LP: Los usuarios de ordenadores de alto rendimiento de las universidades habitualmente tienen cifrado tan solo el acceso a los dispositivos, por lo tanto, trabajan en dispositivos estándar pero con una fuerte criptografía. Lo interesante es que muchos de ellos no tienen ni idea de que lo hace, para el usuario tal técnica

está completamente escondida. SSH es un ejemplo de técnica de cifrado implementada en un entorno IT completamente incoherentemente, y nunca se ha encontrado o sugerido como nociva. No importa desde que lado miremos – la situación parece ideal.

Menos complicado parece el empleo de VPN en el que se introduce el usuario desde casa en la red de la universidad a traves de un túnel no protegido criptográficamente. Primero es necesario construir tal túnel por medio de una aplicación especial de cliente para que los servicios especiales de la universidad se puedan enviar a casa. Ya en este momento muchos usuarios encuentran pequeñas cosas que – para ellos – constituyen un obstáculo importante. Les demostramos a nuestros clientes la necesidad de aplicar los respectivos medios y de tal manera crece la conciencia en este tema. Entonces el acceso a los correos electrónicos en nuestro servidor cifrado solamente es posible con SSL o bien con TLS . Esto se refiere a cada uno de nuestros usuarios y, por lo tanto, todos deben desarrollar una conciencia.

H9: ¿Qué es lo que hacéis concretamente como centro (o tú mismo) para despertar la sensibilidad en este sentido?

LP: Ofrecemos cursillos especiales sobre este tema que dirigimos sobre todo a usuarios noveles. Los cursillos avanzados sobre Linux en general y correo electrónico o sobre el empleo de servidor habitualmente se refieren a este tema paralelamente. Muy importante para los clientes que no quieren o no pueden participar en este tipo de cursillos es la información clara.

Por ejemplo, en nuestro boletín informativo incluimos regularmente artículos o novedades sobre el cifrado de correos o accesos seguros al servidor. Todas las publicaciones estan disponibles para el público en Internet.

H9: ¿La *pereza* es una excusa aceptable? ¿Es difícil para un usuario introducir la criptografía en la práctica?

LP: Sobre todo hay que decir que los métodos criptográficos en el tráfico de correo electrónico son opción decisiva . Está claro que muchas son las causas que favorecen estas prácticas. Cuando uno no quiere, generalmente todo está en orden. La pereza a veces oculta que algo no hacemos bien. Trato de instruir a la gente sobre la situación en la que se encuentran, describir los problemas que encontraron y proponer una solución a estos problemas. Así les conduzco paso a paso a la solución – simplemente introducciendo la firma y cifrado con un clic o una pulsación de tecla. Sin embargo, cuido de que se sigan las reglas básicas lo que hace que el entorno sea transparente y comprensible. En tales condiciones la pereza no es argumento decisivo.

H9: ¿En cuestiones de utilidad ha cambiado mucho últimamenteen cuanto a las aplicaciones de correo electrónico?

LP: Desde hace muchos meses el desarrollo se dirige en el sentido de facilitar el servicio. Buenos ejemplos son los proyectos Thunderbird, Mozilla Mail con Enigmail Plug-in así como el proyecto Kmail que ofrecen excepcionalmente buena integración con las tecnologías de cifrado. Existen muchos clientes de correo electrónico que disponen de las funciones Open PGP y S/MIME. Para muchos entornos gráficos algunos proyectos ofrecen una configuración directa para generar y emplear una clave con GnuPG, por ejemplo, Gnu Privacy Tray bajo Windows Kgpg para KDE. Aquí se trata de aplicaciones muy transparentes y fáciles de soportar, gracias a los cuales el trabajo con código es intuitivo y ligero.

H9: ¿En tu opinión qué es necesario hacer en el trabajo cotidiano si se trata de la seguridad?

LP: Dedicar poco tiempo y dirigirse paso a paso de los problemas a su solución. Los puntos importantes que debemos seguir al cifrar los correos son el tratamiento de las claves, creación de certificados de regreso y el conocimiento de los eslabones más débiles en la cadena de medios que garantizan la seguridad, es decir, las frases de contraseñas para las claves privadas y ellas mismas. Pocas veces escucharemos (y muchas veces se lo recuerdo a los usuarios) que los métodos criptográficos se basan en suposiciones y teoría. Nunca podemos estar completamente seguros. Por ejemplo, cuando encontremos un método rápido de distribuir grandes productos de números primos en los componentes, algunos métodos ya al principio no serán seguros. Hasta el momento en el que los ordenadores cuánticos sean una realidad. Nos olvidamos de que entonces podremos descifrar no solamente los mensajes cifrados sino también todos los mensajes que se creen con este método. Debemos ser conscientes al dar soporte a estas potentes máquinas.

Podemos aprender los principios en unas horas. Excepcionalmente fácil es el soporte para tareas adicionales (Plug-in) de los clientes de correo, integración con SSH, SFTP ó SCP en muchos, muchos productos, por ejemplo, Konqueror (administrador de archivos del proyecto KDE). En los últimos años muchas cosas simplifican y no debemos preocuparnos ya de que haya problemas con el soporte. Cualquiera puede cifrar correos, datos y discos duros completos, quitar el seguro de la comunicación entre ordenadores y muchas más cosas. En general, cuando se entienden los problemas es más fácil solucionarlos. La criptografía en la práctica es mucho menos complicada de lo que suponen los usuarios. Con apenas unos pocos pasos podemos ejecutar ciertas medidas de seguridad y además, convencer de ello a los conocidos.

Una cosa sí es segura: uno puede vernos de muchas formas, así pues la seguridad social, seguros, instituciones estatales y agencias de publicidad se interesarán más. Nosotros, con unos pasos podremos hacer nuestra comunicación más segura.

H9: Gracias por la conversación.

Hablaron: Ulrich Wolf y Heike Jurzik

SOFTWARE PROFESIONAL DE SEGURIDAD

Las bases de datos y la infraestructura que soportan constituyen la fuerza motriz de la organización empresarial. Las bases de datos son las joyas de la corona de una organización (financiera, personal, inventario, procesamiento de tarjetas de crédito etc.). Hay que dar todos los pasos necesarios para corregir todas las Vulnerabilidades de las Bases de Datos. El Escaner de Base de Datos de Safety Lab Shadow es tu ARMA para la DEFENSA.

El escaner de base de datos Safety Lab Shadow suministra todo lo necesario para el análisis y la administración de la base de datos así como para la administración y corrección de vulnerabilidades para la seguridad del servidor SQL. Toda organización dotada de Internet necesita soluciones de absoluta seguridad en las bases de datos sin que éstas dejen de ser flexibles, fáciles de emplear y capaces de guardar recursos de un gran valor.

El escaner de la base de datos Safety Lab cumple con todas estas necesidades, encargandose en las organizaciones de proteger sus datos de gran valor y proteger información para la seguridad del servidor SQL.

¡El escaner de base de datos Shadow (Shadow Database Scanner – escaner de las bases de datos) es una nueva generación de software de alta tecnología que desempeñó su función excepcionalmente en el siglo XX y sigue estando en la línea de frente durante el nuevo milenio!

Shadow Database Scanner ha sido desarrollado para suministrar una segura, súbita y perfecta detección de un amplio rango de huecos y fallos del sistema de seguridad. Después de completar el escaneo del sistema, Shadow Database Scanner analiza los datos recogidos, localiza las vulnerabilidades y posibles errores en la regulación de las opciones del servidor y sugiere posibles formas de solucionar problemas. Shadow Database Scanner emplea un único algoritmo de análisis de la seguridad del sistema basado en patentado "núcleo intelectual".

Gracias a una única arquitectura, Shadow Database Scanner es el único escaner de seguridad del mundo capaz de detectar fallos en MiniSql. Es el único escaner comercial capaz de realizar más de 300 controles por sistema.

Actualmente, soporta los siguientes servidores SQL: MSSql, Oracle, IBMDB2, MiniSql, MySQL, Sybase, SAP DB y Lotus Domino. Gracias a la completamente abierta (a base de ActiveX) arquitectura, cualquier profesional con conocimiento de VC++, C++ Builder o Delphi puede fácilmente desarrollar las capacidades del Escaner.

La tecnología ActiveX también permite a los administradores del sistema integrar Shadow Database Scanner en un producto que prácticamente soporte cualquier ActiveX.

Como Shadow Database Scanner suministra un acceso directo a su núcleo, puedes emplear el API (para una información más detallada, por favor, acudan a la documentación de su API) para conseguir control completo de Shadow Database Scanner o cambiar sus propiedades y funciones. Si no eres programador profesional pero tienes conocimientos básicos de las redes de ordendadores y has encontrado una nueva violación de seguridad puedes también ponerte en contacto directamente con el asistente de Safety-Lab BaseSDK: este te guiará por todo el proceso de creación de un nuevo control. BaseSDK también permite añadir más de un 95% de nuevos tipos de control.

El editor de reglas y configuración serán el punto más importante para los usuarios que quieran solamente escanear los puertos deseados y servicios sin gastar su tiempo y recursos en escanear otros servicios.

La regulación flexible permite a los administradores del sistema administar la profundidad de escaneo y otras opciones para aprovechar del escaneo de red optimizado para su velocidad sin que se produzca ninguna pérdida en la calidad del escaneo.

Otra capacidad única del Escaner se refiere a la posibilidad de guardar registro detallado de una sesión de escaneo no solamente en el formato tradicional HTML (que es accesible en un 99% de otros escaneres) pero también en los formatos XML, PDF, RTF y CHM (compilado HTML).

La nueva interfaz resulta familiar para el usuario, es simple de emplear y ha sido optimizada para suministrar incluso un acceso más fácil a las funciones principales de la aplicación.

La administración de las opciones de Shadow Database Scanner es también más simple: todos los elementos claves de la interfaz de la aplicación tienen ventanas de ayuda en forma de globos con una detallada descripción de sus funciones.

El Asistente de Actualización suministra las actualizaciones provisionales de los módulos ejecutivos de la aplicación con la información más nueva de la seguridad más actual, garantizando una sólida protección para tu sistema y una alta resistencia a los ataques maliciosos.

Safety – Lab también ha incluido en su nuevo producto el acceso directo a su servicio de Internet Security Expert y a la Zona de Descargas actualizada al día.

Si tienes cualquier duda o quieres preguntar sobre los precios para los compradores mayoristas/vendedores de software o tienes una propuesta de negocio, por favor, ponte en contacto con Edward Fitzgerald en la dirección *Edward@safety-lab.com*. ●

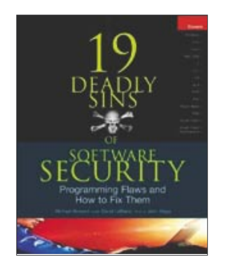
Contacto: Safety-Lab
http://www.safety-lab.com/en

¡Atención!

Safety Lab ofrece a los lectores de *hakin9* la versión completa de Shadow Database Scanners para 2 direcciones IP y con una duración de 30 días.

Al enviar un correo electrónico a *support@safety-lab.com*, debes introducir 2 direcciones IP del Servidor de las Bases de Datos.

La versión completa será válida durante los 30 días a partir de la fecha de recepción de las direcciones IP. Para recibirla, por favor, ponte en contacto a *support@safety-lab.com* escribiendo en en asunto del email hakin9-safety-lab-offer. Válida hasta el 30 de Septiembre de 2006.



Titulo: 19 Deadly Sins of Software Security. Programing Flows and How to Fix Them
Autor: Michael Howard, David LeBlanc, John Viega
Editorial: MIKOM, *http://mikom.pwn.pl/*

Cada programador, inclusive el principiante, cuyo programa funcione mal aunque sólo sea una vez, sabe lo que significa cometer errores durante la escritura del programa. Sin embargo, basta con leer la descripción de cualquier error crítico posible, para asegurarse qué consecuencias paga la seguridad del sistema por estos fallos. Desbordamiento de búfer, de pila o superación del límite de números ocurren con tan frecuencia que no causan impresión en casi nadie.

19 pecados mortales del título, según los autores del libro son unos fallos que se cometen con más frecuencia. En opinión de Amit Yoran (National Cyber Security Division) estos errores representan el 99% de los defectos de software.

Es fácil de adivinar que el libro está dividido en 19 capítulos. Cada uno de ellos se dedica a uno de pecados del título, que podemos cometer escribiendo los programas. Cada capítulo contiene la descripción del *pecado*, un fragmento sobre pecados similares en que se describe las semejanzas entre varios errores y naturalmente está también *la promesa de mejora*; es decir, un párrafo en que se comenta los modos de evitar un error determinado y *examen de conciencia*, que es una relación de problemas y cuestiones que hay que pensar para evitar este pecado en un futuro.

El convencionalismo de tratar el análisis de errores como una confesión peculiar, al principio puede sorprender e incluso irritar, ya que a nadie le gusta admitir que haya cometido un error. Sin embargo, después de meditarlo, hay que admitir que este modo de considerar el asunto es bastante divertido y lógico.

Los autores consiguieron crear una obra rica en contenido, sin notas inútiles e innecesarias – evitaron cometer el pecado más frecuente entre los autores de la literatura informática:

el pecado de escribir una *narración*, así como también cumplieron con la promesa hecha al principio, que el lector recibe una publicación que no le robe el tiempo.

Por eso, programador, no importa que te consideres avanzado: lee, date golpes de pecho y haz penitencia … perfeccionando tu código.



Titulo: Network Security Bible
Autor: Eric Cole, Ronald L. Kurtz, James Conley
Editorial: Helion, *http://www.helion.pl/*

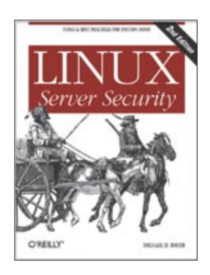
La editorial Helion, en su serie *bíblica* intenta seleccionar unos libros que sean el compendio más completo y extenso de conocimientos sobre un tema. Hasta ahora, por lo menos en lo que se trata a los libros relacionados con la seguridad, la editorial siempre ha conseguido un buen resultado. Sin embargo, los lectores ya se han acostumbrado que esta serie está dirigida a unos usuarios medio avanzados y avanzados.

La seguridad de la red es exactamente un libro escrito para los usuarios del nivel medio avanzado. Ofrece una vasta descripción del problema del título y el lector puede reflexionar sobre lo que ha descuidado durante el proceso de protección de la red. Además, es un libro bien escrito con unas pautas necesarias relativas a la construcción de la Red, un libro en que hay una relación de asuntos esenciales tanto a tiempo de construir una nueva red, así como también proteger la red ya existente.

Al principio aplacamos el entusiasmo de los posibles lectores: la seguridad de la red es un problema tan extenso que una publicación de más de 600 páginas, no puede tratar el tema de modo exhaustivo. Es más, los ejemplos de soluciones y de técnicas presentadas en el libro en principio son superficiales: sirven sólo para ejemplificar las cuestiones escogidas, no obstante muchos problemas sólo han sido mencionados superficialmente.

Por eso, a los usuarios interesados en un tema determinado, no les queda otro remedio que buscar las descripciones de posibles soluciones en otra publicación.

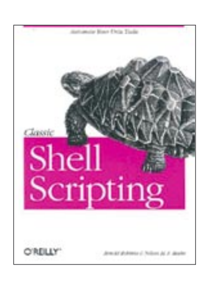
No obstante, la ventaja del libro comentado consiste en recoger en el primer capítulo los procedimientos y las normas generales relativas a la seguridad de la red. Los conocimientos de la seguridad evolucionan de modo muy rápido, sin embargo los procedimientos cambian pocas veces. Los administradores olvidan a menudo que la seguridad de la red consiste también en la seguridad de las informaciones, pues no es sólo la tecnología, sino que también un conjunto de procedimientos, normas y principios.



Titulo: Linux. Server Security.
Autor: Michael D. Bauer
Editorial: Helion, *http://www.helion.pl/*

Éste es un libro que nos puede sorprender antes de leerlo. Por inadvertencia, podemos salir de la librería con un libro anterior de este autor, publicado en el año 2003, por causa de su portada asombrosamente parecida y el título que puede introducir en error: *Linux. Seguridad de servidores*. Sin embargo, si compramos el libro debido, nos sorprenderemos otra vez.

Compramos un libro con una estructura poco homegénea y con un nivel desigual. Del título resulta que es un libro dirigido a un círculo de lectores avanzados con mucha experiencia, que no necesitan un conjunto de indicaciones sobre la configuración de servicios neurálgicos desde el punto de vista de la seguridad, sino las pautas cómo y en qué lugar se puede descuidar algo o cómo perfeccionar unas solucionaes. Pero resulta que se nos ofrece una publicación en que los fragmentos con las consideraciones teóricas (p.e. sobre la ideología del funcionamiento de LDAP) se alternan con las descripciones detalladas del modo de configurar unos servicios determinados. Luego, se nos presenta ni más ni menos, una receta de configuración de unos programas particulares, como Sendmail o Postfix. Todo esto se entrelaza con unos fragmentos flojos, en que el autor se limita casi a constatar que un servicio determinado existe y que puede originar los problemas (por ejemplo las bases MySQL). Por eso, después de leer el libro es difícil juzgar inequívocamente sobre su nivel, o indicar el grupo de las personas a las que está destinado. Sólo se puede afirmar que el libro indudablemente será muy útil para los administradores principantes, aunque sea por causa de las pautas muy detalladas acerca de la configuración FTP, servicio DNS o SMTP. Les servirá también una configuración ejemplar IPTables, o el capítulo sobre los IDS más populares de tipo Snort o Tripware, en que se presenta muchos detalles. Sin embargo, los usuarios más avanzados también van a encontrar algo interesante, bajo la condición de que pacientemente pasen por alto las descripciones de bastantes obviedades.



Titulo: Classic Shell Scripting
Autor: Arnold Robbins, Nelson H. F. Beebe
Editorial: Helion, *http://www.helion.pl/*

Los intérpretes de comandos tratan de la programación en consola de texto, de falta aparente de facilidades y de una técnica flexible para resolver los problemas, tanto de cada día, como de los problemas más peculiares.

Las normas de activar, cerrar o controlar el desarrollo del funcionamiento del sistema, en la mayoría son unos scripts muy complejos. Por eso, para dominarlas, hay que saber operar sobre ellas.

Los comandos del intérprete ofrecen también vastas posibilidades de realizar unas tareas específicas, para las que todavía nadie haya escrito una herramienta apropriada.

Por eso, el libro presentado será de gran utilidad no sólo para los administradores de sistemas de Unix, sino que también para otros usuarios que desean elaborar con facilidad sus propias soluciones, sin necesidad de utilizar los lenguajes de programación típicos.

El libro ha sido escrito de modo transparente y ofrece al lector la posibilidad de conocer las cuestiones cada vez más complejas, relacionadas con la programación de scripts. En los capítulos sucesivos se describe las maneras de hacer uso de los intérpretes de comandos y de sus comandos básicos.

La publicación puede ser tratada como un conjunto de clases, en que los autores no sólo describen las herramientas, sus ventajas, defectos o rango de funcionamiento, sino que también enseñan al lector cómo crear y modificar los scripts hechos. Aconsejan modificarlos de acuerdo con el refrán "No hagas inútilmente algo que ya ha sido hecho" – si alguien ya ha construido algo similar, basta con modificar una parte o un fragmento para crear una solución nueva y adaptable, y a la vez mantener los estándares. El libro puede tener un gran valor tanto para los programadores, como para los administradores que quieren aprender algo sobre la programación de scripts.

Las reseñas de los libros han sido escritas por Krystyna Wal y Łukasz Długosz del grupo InfoProf (*http://www.infoprof.pl/*).

Los libros han sido facilitados a los autores por la Librería Informática en Cracovia (*http://www.informatyczna.pl/*), y a la redacción por las editoriales Helion y WNT.



Folletín

¿Por qué no hay anti-virus?

Konstantin Klyagin 

Si existiera un Greenpeace para la seguridad informática, que cuidara de las especies raras en el mundo de la tecnología de la información, nunca dejaría que se extinguiera una criatura muy interesante. Criatura que celebra este año su 20 aniversario.

Estoy hablando del primer virus para PC, llamado Brain. Aunque no es el primer virus que existió jamás (que apareció antes, en el año 1982 para la plataforma Apple II), el virus Brain es importante para el mundo de los ordenadores, como lo es Brian Adams para su sello discográfico. Cuando el PC fue inventado, los ordenadores empezaron a jugar un papel importante en la vida de la gente normal, no sólo en la de concienzudos científicos. De esta manera los Virus de PC se convirtieron en fenómeno importante.

Otro producto de software para la plataforma Intel, creado en 1985, justo un año después que Brain, ha sobrevivido exitosamente hasta nuestros días. Inicialmente un entorno gráfico para jugar al solitario, al principio no escribía nada de su código en el sector de arranque – corría bajo el sistema operativo MS-DOS. Ahora, después de 21 años en desarrollo y sucesivas versiones, Windows es un sistema operativo en si mismo. Este año una nueva versión, llamada Vista, va a ser publicada. Microsoft no va a incluir ningún software de anti-virus en el paquete, aunque la principal nueva característica de Vista sea la seguridad.

Dudo que esa sea la verdadera razón para no incluir por defecto un anti-virus en el paquete de la distribución. Si se lo permitieran, esos chicos incluirían un aspirador en el sistema operativo simplemente para convertirte en un usuario de tecnología Microsoft. Y no sería demasiado malo. Cuanto más pienso sobre ello, más atractiva me parece la idea. No, en serio, ¿Para qué buscar un aspirador de otra marca si ya tengo el de mi sistema operativo favorito? No te olvides de que una versión OEM cuesta alrededor de 60 €, añadido al precio de mi portátil.

De cualquier manera, ¿Por qué no hay antivirus? La pregunta es fácil de responder, especialmente si recordamos las batallas jurídicas de la compañía de Bill Gates por Internet Explorer y después por Windows Media Player. En ambos casos hubo problemas con el hecho de que las aplicaciones fueran incluidas con el sistema operativo. Con razón no quieren repetirlo con el anti-virus.

Pensemos en los consumidores que realmente pagan por los productos de Microsoft. Mayoritariamente son empresas o usuarios de productos OEM, que compran Windows junto con sus ordenadores. Supongo que no tienen ningún problema en conectar sus equipos a Internet. Además, suelen estar permanentemente conectados.

Hablando claramente, es suficiente con añadir un botón de descargar anti-virus al panel de control para incluirlo virtualmente en la distribución. Un simple click, y ya está, instalado y corriendo.

Symantec y McAfee ya han declarado que aceptarían a Microsoft como un nuevo competidor en el mercado de anti-virus y no llamarán a los legisladores para que hagan algo al respecto. Está vez todo está claro,el software no está incluido, por lo tanto no se aplica ninguna ley antitrust.

Si fuera arquitecto de software en Microsoft, diseñaría el anti-virus para Vista de la siguiente manera. Para evitar un largo tiempo de descarga, pondría todo el código del anti-virus en la librearía de sistema del SO. Lo que se descarga realmente cuando aprietas el botón es un EXE de 5 Kilobytes, que simplemente llama a la función del API `StartAntivirus()`. Ningún problema con la API, ya que de cualquier forma el código es cerrado. Una solución limpia.

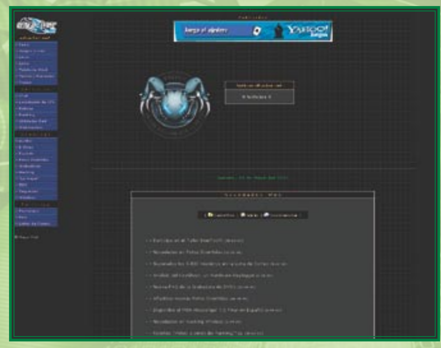
Ahora, ¿Qué pueden hacer Symantec, McAfee y otros vendedores pequeños de anti-virus con respecto a esto? Hay muchas opciones para ellos. Las prospecciones en otros mercados son realmente buenas. Los aspiradores no son la peor opción. ●

Sobre el Autor

Konstantin Klyagin, también conocido como Konst, es un ingeniero de software que lleva 7 años desarrollando software. A los 24, tenía alrededor de 16 años de experiencia con ordenadores, Master en matemáticas aplicadas, habla ruso, ingles, rumano y ucraniano. Nacido en Kharkov, Ucrania, actualmente vive en Berlin.

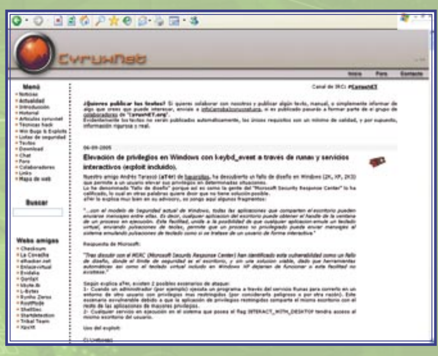
Más información en: *http://thekonst.net/.*

Páginas recomendadas



Una especie de portal para la gente a que le gusta la informática y la seguridad. Si te gusta este mundo, te gustará elhacker.net.

http://www.elhacker.net



CyruxNET – allí encontrarás la información detallada sobre las técnicas hack más populares.

http://www.cyruxnet.org



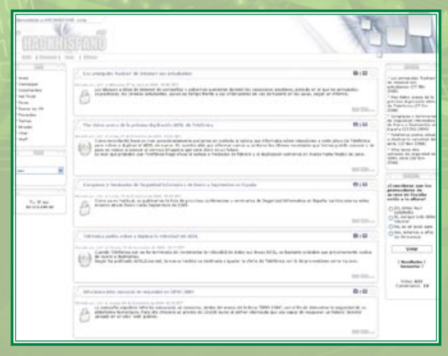
Sitio de noticias que brinda la más variada información en cuanto al mundo de los móviles, enlaces, contactos, y mucho más.

www.diginota.com



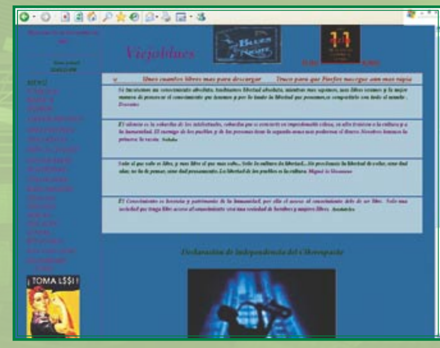
Un lugar de encuentro para todos interesados en temas de seguridad

www.daboweb.com



Hack Hispano, comunidad de usuarios en la que se tratan temas de actualidad sobre nuevas tecnologías, Internet y seguridad informática.

http://www.hackhispano.com



Un espacio libre para compartir: descargas, software, programas oscuros, dudas, noticias, trucos... y más cosas a ritmo de blues.

http://www.viejoblues.com



Aqui encontraras todo lo que debes saber

www.segu-info.com.ar



Tecnología, informática e Internet. Allí encontrarás enlaces, foros, fondos de escritorio y una biblioteca repleta de artículos interesantes...

http://www.hispabyte.net



Indaya teaM fue creada por un grupo de personas amantes de la informática para ayudar a todos los que se interesan por la informática.

http://www.indaya.com



Web especializada en artículos técnicos sobre Linux. Aquí encontrarás las últimas noticias sobre Linux y Software Libre, foros.

www.diariolinux.com



Seguridad0 es un magazine gratuito de seguridad informática. Tiene una periodicidad semanal, aunque se anaden noticias a diario.

http://www.seguridad0.com



DelitosInformaticos.com revista digital de información legal sobre nuevas tecnologías.

www.delitosinformaticos.com

Páginas recomendadas

Si tienes una página web interesante y quieres que la presentemos en nuestra sección de "Páginas recomendadas" contáctanos: *es@hakin9.org*



hakin9 5/2006
En el número siguiente, entre otros:



Criptografía Resistente



Siempre que la gente escribe o habla acerca de la criptografía suelen citar el ejemplo de que el Email es tan seguro como escribir una postal. La criptografía no sólo permite hacer tus comunicaciones por Internet más seguras y confidenciales dando la oportunidad de encriptar o firmar los mensajes. Lars Packschies afirma que la criptografía te da algo como la privacidad en la era de la información y que se está haciendo más y más importante en nuestros días.



Escaneo de puertos y violación de derechos



La propiedad (definida en términos legales) en lo que se refiere a servidores, routers y sistemas de información en general está formada por bienes muebles. Los servidores son bienes muebles. Los datos son de propiedad intelectual. Craig S. Wright nos da una perspectiva interesante sobre el escaneo de puertos y la violación de derechos.



Correlación de sucesos con Simple Event Correlator (SEC) para monitorización en tiempo real



En lo referente a la seguridad de un sistema IT, los registros de sucesos juegan un papel crucial. Hoy en día, muchas aplicaciones, sistemas operativos, dispositivos de red y otros componentes son capaces de escribir mensajes de sucesos relacionados con la seguridad en los archivos de registro. Risto Vaarandi nos muestra todas las funcionalidades del protocolo syslog de BSD, que es un estándar de registro de sucesos soportado por la mayoría de los sistemas operativos y vendedores de dispositivos de red, que nos permite crear un servidor central de registros que reciba y almacene todos los mensajes de sucesos para todo el sistema IT.



Análisis Diferencial de Firewalls



Cómo es posible detectar una violación de las reglas de un firewall usando un sistema de detección de intrusiones de red, comparando en tiempo real el tráfico del exterior con el del interior y viendo si está contradiciendo las reglas. Arrigo Triulzi y Antonio Merola discutiremos como un Sistema de Detección de Intrusiones de red (Network Intrusion Detection System – NIDS) puede usarse como herramienta de verificación en el caso específico de un fallo de firewall.

Información actual sobre el próximo número – *http://www.hakin9.org/es*

El número está a la venta desde principios de Septiembre de 2006.

La redacción se reserva el derecho a cambiar el contenido de la revista.